Spazi Vettoriali Semantici: Geometria del Significato nei Word Embeddings

Introduzione: La Rivoluzione della Rappresentazione Distribuita del Significato

La comprensione del linguaggio naturale da parte delle macchine ha rappresentato per decenni una delle sfide più ardue dell'intelligenza artificiale. Per lungo tempo, i sistemi computazionali hanno trattato le parole come simboli discreti e atomici, privi di struttura interna e di relazioni significative tra loro. Questa visione simbolica del linguaggio, sebbene intuitiva da un punto di vista formale, si è rivelata profondamente inadeguata nel catturare la ricchezza e la complessità delle relazioni semantiche che caratterizzano il linguaggio umano. La svolta concettuale è arrivata con l'introduzione degli spazi vettoriali semantici, un paradigma che ha trasformato radicalmente il modo in cui rappresentiamo e processiamo il significato linguistico attraverso rappresentazioni geometriche continue.

Gli spazi vettoriali semantici costituiscono un framework matematico nel quale ogni parola viene rappresentata come un vettore in uno spazio multidimensionale, tipicamente caratterizzato da centinaia o migliaia di dimensioni. Questa rappresentazione, nota come word embedding, non è arbitraria ma emerge da processi di apprendimento che analizzano enormi quantità di testo, catturando le regolarità statistiche della distribuzione delle parole nei contesti linguistici. Il principio fondamentale che governa questa costruzione è l'ipotesi distribuzionale, formulata dal linguista John Rupert Firth negli anni Cinquanta del secolo scorso con la celebre massima: "You shall know a word by the company it keeps". Secondo questa ipotesi, il significato di una parola è definito dai contesti in cui essa appare, e parole che occorrono in contesti simili tendono ad avere significati simili.

Ciò che rende gli spazi vettoriali semantici straordinariamente potenti è la capacità di codificare relazioni semantiche complesse attraverso operazioni geometriche elementari. La distanza euclidea, il prodotto scalare, gli angoli tra vettori diventano strumenti per misurare la similarità semantica, per identificare sinonimi, per catturare relazioni analogiche. In questi spazi emerge una geometria del significato che riflette in modo sorprendentemente accurato le intuizioni linguistiche umane. La celebre equazione vettoriale "re - uomo + donna ≈ regina" non è una curiosità matematica, ma un esempio emblematico di come le relazioni concettuali complesse possano essere codificate e manipolate attraverso semplici operazioni algebriche nello spazio vettoriale.

La storia degli word embeddings è intimamente legata allo sviluppo delle reti neurali e dell'apprendimento automatico. Sebbene le prime intuizioni risalgano agli anni Ottanta e Novanta, con i modelli connessionisti di rappresentazione distribuita proposti da Geoffrey Hinton, David Rumelhart e altri pionieri del connessionismo, la vera esplosione di interesse è arrivata nei primi anni del ventunesimo secolo. Nel 2003, Yoshua Bengio e il suo gruppo di ricerca hanno introdotto il modello di linguaggio neurale, dimostrando che le reti neurali

potevano apprendere rappresentazioni dense e continue delle parole come sottoprodotto dell'addestramento per predire sequenze linguistiche. Questo lavoro ha aperto la strada a una serie di sviluppi che hanno rivoluzionato il campo del processamento del linguaggio naturale.

Nel 2013, Tomas Mikolov e i suoi collaboratori presso Google hanno introdotto Word2Vec, un insieme di architetture e tecniche di addestramento che hanno reso gli word embeddings accessibili e praticamente utilizzabili su larga scala. Word2Vec ha democratizzato l'uso degli embeddings, fornendo implementazioni efficienti e pre-addestrate che hanno permesso a ricercatori e sviluppatori di tutto il mondo di sfruttare rappresentazioni vettoriali di alta qualità senza la necessità di risorse computazionali enormi. Parallelamente, altri approcci come GloVe, sviluppato da Jeffrey Pennington, Richard Socher e Christopher Manning all'Università di Stanford, hanno esplorato metodi alternativi basati sulla fattorizzazione di matrici di co-occorrenza, dimostrando che esistono molteplici vie per arrivare a rappresentazioni vettoriali semanticamente ricche.

Gli word embeddings non sono semplicemente una tecnica tra le tante nel panorama del machine learning, ma rappresentano un cambio di paradigma fondamentale nella nostra comprensione di come le macchine possano processare e "comprendere" il linguaggio. Essi hanno reso possibile lo sviluppo di sistemi di traduzione automatica di qualità senza precedenti, di assistenti conversazionali sempre più sofisticati, di motori di ricerca semantica che vanno oltre il semplice matching di parole chiave. Ma soprattutto, gli word embeddings hanno aperto una finestra affascinante sulla struttura del significato linguistico, rivelando pattern e regolarità che erano rimasti nascosti nelle analisi linguistiche tradizionali.

Il fascino particolare degli spazi vettoriali semantici risiede nella loro capacità di fare emergere proprietà semantiche complesse in modo completamente non supervisionato. Gli algoritmi che generano embeddings non ricevono informazioni esplicite su quali parole siano sinonimi, quali siano antonimi, quali relazioni analogiche esistano tra i concetti. Eppure, analizzando semplicemente la distribuzione statistica delle parole nei testi, questi algoritmi riescono a costruire rappresentazioni geometriche nelle quali le relazioni semantiche si manifestano come pattern geometrici regolari e interpretabili. Questo fenomeno suggerisce che la struttura del significato linguistico non è arbitraria ma riflette principi organizzativi profondi che emergono dall'uso comunicativo del linguaggio.

La dimensionalità degli spazi vettoriali semantici merita una riflessione particolare. Mentre il vocabolario di una lingua può contenere centinaia di migliaia o milioni di parole distinte, gli embeddings tipicamente utilizzano rappresentazioni con dimensioni che variano da poche decine a diverse migliaia. Word2Vec, per esempio, utilizza comunemente vettori di 100, 200 o 300 dimensioni. Questa riduzione drastica della dimensionalità rispetto alla rappresentazione one-hot tradizionale, nella quale ogni parola è rappresentata da un vettore con tante dimensioni quante sono le parole nel vocabolario, non è una perdita di informazione ma al contrario una forma di compressione intelligente che cattura le regolarità sottostanti del linguaggio. Le dimensioni dello spazio vettoriale possono essere interpretate come direzioni astratte che codificano aspetti del significato, anche se queste dimensioni raramente corrispondono a categorie semantiche interpretabili direttamente dall'uomo.

L'emergere della geometria del significato negli spazi vettoriali semantici solleva questioni profonde sulla natura della rappresentazione linguistica e cognitiva. Le rappresentazioni distribuite sembrano catturare aspetti del significato che sfuggono alle analisi simboliche tradizionali, suggerendo che il significato non sia una proprietà atomica e discreta ma piuttosto un fenomeno emergente che deriva dalla posizione di un concetto in una rete di relazioni con altri concetti. Questa visione risuona con teorie linguistiche e cognitive che enfatizzano la natura relazionale e contestuale del significato, ma la formalizza in un framework matematico rigoroso che permette manipolazioni e predizioni quantitative.

Fondamenti Matematici: La Struttura Algebrica degli Spazi Vettoriali

Per comprendere appieno come gli spazi vettoriali semantici codificano il significato linguistico, è essenziale padroneggiare i fondamenti matematici che sottendono questa rappresentazione. Uno spazio vettoriale è una struttura algebrica definita da un insieme di vettori e da due operazioni fondamentali: l'addizione vettoriale e la moltiplicazione per uno scalare. Formalmente, uno spazio vettoriale V su un campo scalare F, tipicamente i numeri reali R, è un insieme dotato di un'operazione di addizione + : V × V → V e di un'operazione di moltiplicazione scalare · : F × V → V che soddisfano una serie di assiomi.

Gli assiomi che definiscono uno spazio vettoriale includono la commutatività e l'associatività dell'addizione vettoriale, l'esistenza di un elemento neutro per l'addizione chiamato vettore zero, l'esistenza di un inverso additivo per ogni vettore, la distributività della moltiplicazione scalare rispetto all'addizione vettoriale e all'addizione di scalari, e l'associatività della moltiplicazione scalare. Formalmente, dati vettori u, v, w ∈ V e scalari a, b ∈ F, devono valere le seguenti proprietà: u + v = v + u, (u + v) + w = u + (v + w), esiste 0 ∈ V tale che v + 0 = v per ogni v, per ogni v esiste -v tale che v + (-v) = 0, a(u + v) = au + av, (a + b)v = av + bv, (ab)v = a(bv), e 1v = v dove 1 è l'unità del campo scalare.

Nel contesto degli word embeddings, lo spazio vettoriale è tipicamente R^d, lo spazio euclideo d-dimensionale, dove d rappresenta la dimensionalità degli embeddings. Ogni parola w nel vocabolario è associata a un vettore v_w ∈ R^d, dove v_w = (v_w1, v_w2, ..., v_wd) e ogni componente v_wi è un numero reale. La scelta della dimensionalità d è un iperparametro cruciale che bilancia capacità rappresentativa e efficienza computazionale. Dimensioni troppo basse possono non catturare adeguatamente le sfumature semantiche, mentre dimensioni troppo alte possono portare a sovradattamento e inefficienza computazionale.

La norma di un vettore è una funzione che assegna a ogni vettore una misura della sua "lunghezza" o "magnitudine". La norma euclidea, anche chiamata norma L2, è definita come ||v|| = sqrt(v1^2 + v2^2 + ... + v_d^2). Questa norma induce una metrica, la distanza euclidea, che misura la distanza tra due vettori come ||v - w|| = sqrt(sum_{i=1}^d (v_i - w_i)^2). La distanza euclidea è la metrica più naturale e comunemente utilizzata per misurare la dissimilarità semantica tra parole negli spazi vettoriali. Parole semanticamente simili tendono ad avere vettori vicini nello spazio, con distanza euclidea piccola.

Il prodotto scalare, o prodotto interno, è un'operazione fondamentale che prende due vettori e restituisce uno scalare. Il prodotto scalare standard in R^d è definito come v · w = sum_{i=1}^d v_i w_i = v1 w1 + v2 w2 + ... + v_d w_d. Il prodotto scalare è strettamente legato alla norma attraverso la relazione ||v||^2 = v · v e alla nozione di angolo tra vettori. L'angolo θ tra due vettori non nulli v e w è definito dalla relazione cos(θ) = (v · w) / (||v|| ||w||). Questa relazione è fondamentale per definire la similarità del coseno, una delle misure più utilizzate negli spazi vettoriali semantici.

La similarità del coseno tra due vettori v e w è definita come cos_sim(v, w) = (v · w) / (||v|| ||w||) = sum_{i=1}^d v_i w_i / (sqrt(sum_{i=1}^d v_i^2) sqrt(sum_{i=1}^d w_i^2)). Questa misura varia tra -1 e 1, dove 1 indica vettori perfettamente allineati nella stessa direzione, 0 indica vettori ortogonali senza relazione direzionale, e -1 indica vettori opposti. La similarità del coseno è particolarmente utile perché è invariante rispetto alla scala: normalizza i vettori alla lunghezza unitaria prima di misurare l'angolo, catturando così la somiglianza direzionale indipendentemente dalla magnitudine. Negli spazi vettoriali semantici, la similarità del coseno è spesso preferita alla distanza euclidea perché cattura meglio l'intuizione che parole semanticamente simili dovrebbero puntare nella stessa direzione nello spazio, indipendentemente da quanto siano "frequenti" o "salienti".

L'ortogonalità è un concetto geometrico che ha implicazioni semantiche interessanti. Due vettori v e w sono ortogonali se il loro prodotto scalare è zero: v · w = 0. Geometricamente, questo significa che formano un angolo di 90 gradi. Negli spazi vettoriali semantici, vettori ortogonali rappresentano concetti semanticamente indipendenti o non correlati. Tuttavia, è importante notare che l'assenza di correlazione geometrica non implica necessariamente l'assenza totale di relazione semantica, poiché le relazioni linguistiche sono complesse e multidimensionali.

Le operazioni vettoriali di addizione e sottrazione acquisiscono un significato semantico particolare negli word embeddings. L'addizione vettoriale v + w combina i significati delle parole corrispondenti, creando una rappresentazione che incorpora aspetti semantici di entrambe. La sottrazione v - w può essere interpretata come l'isolamento della differenza semantica o della relazione direzionale tra due concetti. Queste operazioni sono alla base delle famose analogie vettoriali: se "re" è a "uomo" come "regina" è a "donna", allora il vettore v_re - v_uomo dovrebbe essere approssimativamente uguale a v_regina - v_donna, e quindi v_re - v_uomo + v_donna ≈ v_regina.

La proiezione vettoriale è un'altra operazione geometrica rilevante. La proiezione di un vettore v su un vettore w è il vettore proj_w(v) = ((v · w) / (w · w)) w, che rappresenta la componente di v nella direzione di w. Le proiezioni possono essere utilizzate per estrarre componenti semantiche specifiche o per rimuovere direzioni indesiderate dallo spazio vettoriale, una tecnica nota come debiasing negli studi sulla fairness degli embeddings.

La base di uno spazio vettoriale è un insieme di vettori linearmente indipendenti che generano l'intero spazio. In R^d, la base standard è costituita dai vettori e_1 = (1, 0, 0, ..., 0), e_2 = (0, 1, 0, ..., 0), ..., e_d = (0, 0, 0, ..., 1). Ogni vettore v può essere espresso come combinazione lineare dei vettori di base: v = sum_{i=1}^d v_i e_i. Negli spazi vettoriali semantici, le dimensioni della base non hanno tipicamente interpretazioni semantiche

dirette, ma rappresentano direzioni astratte nello spazio che collettivamente codificano il significato.

Le trasformazioni lineari sono funzioni $T : V \rightarrow W$ tra spazi vettoriali che preservano le operazioni di addizione e moltiplicazione scalare: $T(u + v) = T(u) + T(v)$ e $T(av) = aT(v)$. Le trasformazioni lineari possono essere rappresentate da matrici e sono utilizzate per mappare embeddings tra diversi spazi, per ridurre la dimensionalità, o per apprendere relazioni strutturate tra lingue diverse nella traduzione automatica.

La decomposizione in valori singolari, abbreviata SVD dall'inglese Singular Value Decomposition, è una fattorizzazione di matrici di fondamentale importanza nell'algebra lineare e nelle applicazioni agli embeddings. Data una matrice M di dimensione $m \times n$, la SVD la decompone come $M = U \Sigma V^T$, dove U è una matrice ortogonale $m \times m$, Σ è una matrice diagonale $m \times n$ con valori singolari non negativi sulla diagonale disposti in ordine decrescente, e V^T è la trasposta di una matrice ortogonale $n \times n$. La SVD è alla base di molti metodi di riduzione dimensionale come l'analisi delle componenti principali e viene utilizzata in approcci come la Latent Semantic Analysis per costruire rappresentazioni semantiche.

L'analisi delle componenti principali, abbreviata PCA, è una tecnica di riduzione dimensionale che identifica le direzioni di massima varianza nei dati. Applicata agli embeddings, la PCA può ridurre la dimensionalità preservando la maggior parte dell'informazione semantica, proiettando i vettori su un sottospazio di dimensione inferiore definito dalle componenti principali. Le prime componenti principali catturano le direzioni lungo le quali i vettori di parole variano maggiormente, corrispondendo spesso a distinzioni semantiche salienti.

La geometria degli spazi ad alta dimensionalità presenta fenomeni controintuitivi che hanno implicazioni importanti per gli word embeddings. In dimensioni elevate, i vettori tendono a diventare quasi ortogonali: la maggior parte delle coppie di vettori casuali ha una similarità del coseno vicina a zero. Questo fenomeno, noto come "maledizione della dimensionalità" in alcuni contesti, significa che in spazi ad alta dimensionalità c'è "molto spazio" e i punti possono essere ben separati. Tuttavia, negli embeddings addestrati, le parole non sono distribuite uniformemente ma formano cluster e strutture che riflettono le regolarità semantiche del linguaggio.

Il concetto di convessità è rilevante per comprendere la struttura geometrica degli spazi semantici. Un insieme C è convesso se, per ogni coppia di punti $u, v \in C$, il segmento che li connette è interamente contenuto in C: per ogni $t \in [0, 1]$, il punto $tu + (1-t)v \in C$. Negli spazi vettoriali semantici, i cluster di parole semanticamente correlate tendono a formare regioni approssimativamente convesse, sebbene la convessità esatta non sia garantita. La combinazione convessa di embeddings, ottenuta come media pesata, produce tipicamente vettori che rappresentano concetti intermedi o composizioni semantiche.

Le metriche alternative alla distanza euclidea e alla similarità del coseno includono la distanza di Manhattan, definita come $d_1(v, w) = \sum_{i=1}^{d} |v_i - w_i|$, e la divergenza di Kullback-Leibler per distribuzioni di probabilità. Sebbene la similarità del coseno rimanga la misura predominante, l'esplorazione di metriche alternative può rivelare aspetti diversi della struttura semantica.

La teoria degli spazi metrici fornisce un framework generale per ragionare su distanze e similarità. Uno spazio metrico è un insieme M dotato di una funzione di distanza $d : M \times M \rightarrow R$ che soddisfa le proprietà di non negatività ($d(x, y) \geq 0$), identità ($d(x, y) = 0$ se e solo se $x = y$), simmetria ($d(x, y) = d(y, x)$), e disuguaglianza triangolare ($d(x, z) \leq d(x, y) + d(y, z)$). Gli spazi vettoriali con distanza euclidea sono spazi metrici, e molte tecniche di analisi e indicizzazione si basano su queste proprietà metriche.

L'Ipotesi Distribuzionale: Fondamento Linguistico degli Embeddings

L'ipotesi distribuzionale costituisce il fondamento teorico su cui poggia l'intera architettura degli spazi vettoriali semantici. Questa ipotesi, formulata esplicitamente da linguisti come Zellig Harris e John Rupert Firth a metà del ventesimo secolo, afferma che le parole che occorrono in contesti simili tendono ad avere significati simili. Firth la espresse in modo particolarmente memorabile nel 1957: "You shall know a word by the company it keeps", sottolineando che il significato di una parola è intrinsecamente legato alle parole con cui co-occorre nei testi.

Questa ipotesi rappresenta un cambiamento radicale rispetto alle concezioni tradizionali del significato in semantica formale. Nella semantica vero-condizionale, il significato di un'espressione linguistica è definito dalle sue condizioni di verità, ovvero dalle situazioni nel mondo che renderebbero vera o falsa un'asserzione contenente quell'espressione. Nella semantica strutturalista di Ferdinand de Saussure, il significato è definito dalle relazioni di opposizione e contrasto all'interno di un sistema linguistico. L'ipotesi distribuzionale sposta il focus dalle relazioni con il mondo extralinguistico o dalle relazioni sistematiche astratte alle regolarità osservabili nell'uso linguistico concreto.

La forza dell'ipotesi distribuzionale risiede nella sua operazionalizzabilità computazionale. Mentre nozioni come condizioni di verità o intuizioni semantiche richiedono annotazione umana o accesso a conoscenza del mondo, la distribuzione contestuale delle parole può essere estratta automaticamente da grandi corpora testuali. Questo rende possibile l'apprendimento non supervisionato di rappresentazioni semantiche, un vantaggio cruciale per la scalabilità e l'applicabilità degli word embeddings a lingue e domini diversi.

La validità empirica dell'ipotesi distribuzionale è stata confermata da numerosi studi in psicolinguistica e linguistica computazionale. Parole come "cane" e "gatto", che hanno significati simili in quanto entrambi riferiscono ad animali domestici comuni, tendono effettivamente a occorrere in contesti simili: possono essere precedute da aggettivi come "piccolo", "affettuoso", o "domestico", possono essere soggetti di verbi come "abbaiare" o "miagolare" (sebbene con preferenze diverse), possono apparire in frasi che parlano di cura, alimentazione, o compagnia. Al contrario, parole con significati molto diversi come "cane" e "teorema" appaiono in contesti radicalmente differenti.

Tuttavia, l'ipotesi distribuzionale presenta anche limiti e zone d'ombra che meritano attenzione critica. Un problema fondamentale è quello degli antonimi: parole come "caldo" e "freddo" sono opposte nel significato ma tendono a occorrere in contesti molto simili, descrivendo entrambe proprietà di temperatura e apparendo spesso in frasi strutturalmente

analoghe. Gli embeddings basati puramente sulla distribuzione contestuale possono assegnare alta similarità agli antonimi, catturando la loro affinità topica ma non l'opposizione semantica. Questo suggerisce che la pura co-occorrenza contestuale cattura aspetti importanti ma non esaustivi del significato.

Un altro limite riguarda le parole polisemiche, ovvero parole con molteplici significati correlati. La parola "banco", per esempio, può riferirsi a un istituto finanziario, a un tavolo di lavoro, a un cumulo di sabbia, o a un gruppo di pesci. Gli embeddings tradizionali assegnano un unico vettore a ogni forma di parola, producendo una rappresentazione che mescola tutti i sensi. Sebbene questa rappresentazione aggregata possa essere utile in molte applicazioni, non cattura la struttura interna della polisemia. Approcci più recenti, come i modelli contestualizzati di cui parleremo successivamente, affrontano questo problema generando rappresentazioni diverse per la stessa parola in contesti diversi.

La definizione operativa di "contesto" nell'ipotesi distribuzionale ammette diverse interpretazioni che portano a modelli di embeddings con proprietà diverse. Il contesto può essere definito come una finestra di parole adiacenti di dimensione fissa, come nell'approccio di Word2Vec con il modello Skip-gram e Continuous Bag-of-Words. Oppure può essere definito in termini di dipendenze sintattiche, catturando relazioni grammaticali esplicite. Può anche essere esteso all'intero documento, come nei modelli di topic modeling. Ciascuna di queste operazionalizzazioni cattura aspetti diversi della distribuzione linguistica e porta a embeddings con caratteristiche geometriche diverse.

La teoria dell'informazione fornisce un framework formale per quantificare l'informazione che il contesto fornisce su una parola. L'informazione mutua pointwise, abbreviata PMI, tra una parola w e un contesto c è definita come PMI(w, c) = log(P(w, c) / (P(w) P(c))), dove P(w, c) è la probabilità congiunta di osservare la parola w nel contesto c, P(w) è la probabilità marginale della parola, e P(c) è la probabilità marginale del contesto. La PMI è positiva quando w e c co-occorrono più frequentemente di quanto ci si aspetterebbe se fossero indipendenti, negativa quando co-occorrono meno frequentemente, e zero quando sono statisticamente indipendenti. Molti metodi di embeddings, incluso GloVe, possono essere interpretati come fattorizzazioni di matrici di PMI o di varianti pesate della PMI.

L'ipotesi distribuzionale può essere vista come una forma di empirismo semantico: il significato emerge dall'esperienza linguistica, codificata nelle regolarità statistiche dei testi. Questa visione è coerente con teorie cognitive che enfatizzano il ruolo dell'esperienza e dell'apprendimento nella formazione dei concetti. Tuttavia, solleva anche questioni filosofiche profonde sulla natura del significato e sulla relazione tra linguaggio e mondo. Il significato distribuzionale è sufficiente per catturare tutti gli aspetti rilevanti della semantica linguistica, o è necessario integrarlo con informazioni sul mondo extralinguistico, sulla percezione, sull'azione?

Recenti sviluppi nell'intelligenza artificiale multimodale suggeriscono che integrare informazioni visive, uditive, o di altro tipo può arricchire gli embeddings linguistici. I modelli vision-language come CLIP apprendono rappresentazioni congiunte di testo e immagini, permettendo di ancorare i concetti linguistici a referenti percettivi. Questa integrazione multimodale può essere vista come un'estensione dell'ipotesi distribuzionale che include non solo la distribuzione testuale ma anche la distribuzione cross-modale: il significato di "cane"

è informato non solo dai contesti linguistici ma anche dalle immagini di cani con cui la parola co-occorre.

Word2Vec: Architetture e Meccanismi di Apprendimento

Word2Vec rappresenta probabilmente il punto di svolta più significativo nella storia degli word embeddings, non tanto per l'originalità concettuale assoluta quanto per l'efficacia pratica e l'accessibilità che ha portato a queste tecniche. Introdotto da Tomas Mikolov e colleghi nel 2013 in una serie di lavori pubblicati presso Google, Word2Vec comprende due architetture principali: il modello Skip-gram e il modello Continuous Bag-of-Words, spesso abbreviato come CBOW. Entrambi questi modelli apprendono embeddings addestrandosi a predire parole dal contesto o contesto dalle parole, ma lo fanno in direzioni opposte.

Il modello Skip-gram parte da una parola target e cerca di predire le parole che appaiono nel suo contesto. Formalmente, dato un corpus di parole w_1, w_2, ..., w_T, l'obiettivo dello Skip-gram è massimizzare la log-verosimiglianza media: L = (1/T) sum_{t=1}^T sum_{-c ≤ j ≤ c, j ≠ 0} log P(w_{t+j} | w_t), dove c è la dimensione della finestra di contesto e P(w_{t+j} | w_t) è la probabilità di osservare la parola contestuale w_{t+j} data la parola centrale w_t. Questa probabilità viene parametrizzata usando i vettori di embedding.

Nell'implementazione base di Skip-gram, ogni parola w ha due rappresentazioni vettoriali: un vettore "input" v_w e un vettore "output" u_w, entrambi in R^d. La probabilità di osservare una parola contestuale w_O data una parola centrale w_I è calcolata usando la funzione softmax: P(w_O | w_I) = exp(u_{w_O} · v_{w_I}) / sum_{w ∈ V} exp(u_w · v_{w_I}), dove V è il vocabolario completo. Il prodotto scalare u_{w_O} · v_{w_I} misura la compatibilità tra la parola di output w_O e la parola di input w_I, e la softmax trasforma queste compatibilità in una distribuzione di probabilità normalizzata sull'intero vocabolario.

Il problema computazionale principale di questa formulazione è che calcolare la softmax richiede una somma su tutto il vocabolario, che può contenere centinaia di migliaia o milioni di parole. Per ogni passo di addestramento, questo calcolo diventa proibitivamente costoso. Mikolov e colleghi hanno introdotto due tecniche di approssimazione efficiente: il negative sampling e la softmax gerarchica.

Il negative sampling trasforma il problema di predizione multi-classe in un insieme di problemi di classificazione binaria. Invece di calcolare la probabilità normalizzata su tutto il vocabolario, per ogni coppia parola-contesto positiva (parola effettivamente osservata nel contesto) si campionano k parole negative (parole non osservate in quel contesto) da una distribuzione di rumore. L'obiettivo diventa distinguere le coppie positive dalle coppie negative usando una funzione di loss basata sulla logistica: log σ(u_{w_O} · v_{w_I}) + sum_{i=1}^k E_{w_i ~ P_n(w)} [log σ(-u_{w_i} · v_{w_I})], dove σ(x) = 1 / (1 + exp(-x)) è la funzione sigmoide e P_n(w) è la distribuzione di rumore, tipicamente scelta come una versione modificata della distribuzione unigramma delle parole. La scelta comune è P_n(w) ∝ freq(w)^{0,75}, dove freq(w) è la frequenza della parola w nel corpus, e l'esponente 0,75 aumenta il peso delle parole meno frequenti, migliorando la qualità degli embeddings.

La softmax gerarchica organizza il vocabolario in una struttura ad albero binario, tipicamente un albero di Huffman basato sulle frequenze delle parole. Ogni parola corrisponde a una foglia dell'albero, e la probabilità di una parola viene calcolata come il prodotto delle probabilità binarie lungo il cammino dalla radice alla foglia. Formalmente, ogni nodo interno n dell'albero ha associato un vettore v_n, e la probabilità di andare a sinistra o a destra in quel nodo dato l'input w_I è determinata da σ(v_n · v_{w_I}). La probabilità complessiva di raggiungere una parola w_O è il prodotto delle decisioni binarie lungo il cammino. Questa struttura riduce la complessità del calcolo da O(|V|) a O(log|V|), dove |V| è la dimensione del vocabolario.

Il modello Continuous Bag-of-Words opera nella direzione opposta: dato un contesto, predice la parola centrale. L'architettura CBOW prende i vettori delle parole contestuali, li media, e usa questa rappresentazione aggregata per predire la parola target. Formalmente, dato un contesto di parole w_{t-c}, ..., w_{t-1}, w_{t+1}, ..., w_{t+c}, il modello calcola il vettore medio v_context = (1/(2c)) sum_{-c ≤ j ≤ c, j ≠ 0} v_{w_{t+j}} e poi predice w_t massimizzando P(w_t | v_context) usando una softmax o tecniche di approssimazione simili a quelle dello Skip-gram.

La differenza fondamentale tra Skip-gram e CBOW è nella direzione della predizione e nelle proprietà statistiche. Skip-gram tratta ogni coppia parola-contesto come un esempio di addestramento separato, dando più peso alle parole rare e potenzialmente catturando meglio le sfumature semantiche. CBOW, mediando il contesto, tende ad essere più efficiente computazionalmente e a produrre embeddings leggermente più stabili per le parole frequenti, ma può perdere informazioni su parole rare. In pratica, Skip-gram è generalmente preferito per applicazioni che richiedono embeddings di alta qualità per vocabolari ampi che includono termini meno frequenti.

L'addestramento di Word2Vec utilizza la discesa del gradiente stocastico, aggiornando iterativamente i vettori di embedding per massimizzare l'obiettivo. Il gradiente della funzione obiettivo rispetto ai vettori può essere calcolato usando backpropagation, e gli aggiornamenti vengono applicati dopo aver processato piccoli batch di esempi. La scelta del learning rate, del numero di epoche di addestramento, della dimensione del batch e di altri iperparametri influenza significativamente la qualità degli embeddings risultanti.

Un aspetto cruciale dell'addestramento di Word2Vec è il preprocessing del corpus. Le parole molto frequenti come articoli e preposizioni vengono spesso sottocampionate, riducendo la loro presenza nel training set per bilanciare l'informazione semantica e computare embeddings più informativi per parole di contenuto. Il sottocampionamento avviene con probabilità P(w_i) = 1 - sqrt(t / freq(w_i)), dove t è una soglia (tipicamente intorno a 10^{-5}) e freq(w_i) è la frequenza normalizzata della parola. Questa formula mantiene le parole rare con probabilità vicina a 1 ma riduce drasticamente la presenza di parole molto frequenti.

Word2Vec può essere esteso per gestire frasi e multi-word expressions identificando automaticamente collocazioni frequenti. Due parole consecutive w_i e w_{i+1} vengono combinate in una singola unità se il loro score di collocazione score(w_i, w_{i+1}) = (count(w_i, w_{i+1}) - δ) / (count(w_i) · count(w_{i+1})) supera una soglia, dove count(w_i, w_{i+1}) è il numero di volte che le due parole appaiono consecutive e δ è un fattore di

sconto che previene la formazione di frasi da coppie di parole molto rare. Questa procedura può essere applicata iterativamente per identificare frasi più lunghe.

Gli embeddings di Word2Vec possono essere inizializzati casualmente, tipicamente campionando da una distribuzione uniforme su un intervallo piccolo come [-0,5/d, 0,5/d], dove d è la dimensionalità, o da una distribuzione normale con varianza piccola. L'inizializzazione casuale rompe la simmetria e permette ai vettori di differenziarsi durante l'addestramento. Tuttavia, l'inizializzazione può influenzare la convergenza e la qualità finale, specialmente per parole rare che ricevono pochi aggiornamenti.

La funzione di loss di Word2Vec, sia con negative sampling che con softmax gerarchica, è non convessa, il che significa che possono esistere molteplici minimi locali. L'addestramento con discesa del gradiente stocastico non è garantito trovare l'ottimo globale, ma in pratica converge a soluzioni di alta qualità. La stocasticità nell'ordine di presentazione degli esempi e nell'inizializzazione agisce come regolarizzazione implicita, aiutando a evitare overfitting.

Una variante interessante è l'uso di contesti sintattici invece di contesti basati su finestre di parole adiacenti. Invece di considerare tutte le parole entro una finestra di c posizioni, si possono considerare solo parole connesse da specifiche relazioni sintattiche come soggetto-verbo, verbo-oggetto, modificatore-nome. Questo approccio, proposto da Levy e Goldberg, produce embeddings che catturano maggiormente similarità funzionale e sintattica piuttosto che similarità topica. Per esempio, "cane" e "gatto" potrebbero essere meno simili con contesti sintattici rispetto a contesti di finestra, mentre "cane" e il verbo "abbaiare" potrebbero essere più correlati.

GloVe: Fattorizzazione di Matrici di Co-occorrenza Globale

GloVe, acronimo di Global Vectors for word representation, rappresenta un approccio complementare agli word embeddings che enfatizza l'importanza delle statistiche di co-occorrenza globale del corpus. Sviluppato da Jeffrey Pennington, Richard Socher e Christopher Manning all'Università di Stanford e pubblicato nel 2014, GloVe parte dall'osservazione che i metodi basati su finestre locali come Word2Vec utilizzano informazioni contestuali locali ma non sfruttano esplicitamente le statistiche aggregate di co-occorrenza che emergono dall'intero corpus.

L'idea centrale di GloVe è che i rapporti di probabilità di co-occorrenza, piuttosto che le probabilità stesse, codificano informazioni semantiche distintive. Consideriamo tre parole: "ghiaccio", "vapore", e una terza parola contestuale k. Se k è "solido", ci aspetteremmo che co-occorra più frequentemente con "ghiaccio" che con "vapore". Se k è "gas", ci aspetteremmo l'opposto. Se k è "acqua", potrebbe co-occorrere similarmente con entrambe, mentre se k è completamente non correlata come "moda", co-occorrerebbe raramente con entrambe. Il rapporto P(k | ghiaccio) / P(k | vapore) cattura queste distinzioni in modo più discriminativo rispetto alle probabilità individuali.

GloVe costruisce una matrice di co-occorrenza X dove X_{ij} rappresenta il numero di volte che la parola j appare nel contesto della parola i. La definizione di contesto è flessibile: può

essere una finestra simmetrica intorno alla parola, con possibile pesatura decrescente per parole più distanti. Gli elementi della matrice possono essere conteggi grezzi o pesati. GloVe poi cerca di apprendere vettori di parole v_i e vettori di contesto u_j tali che il loro prodotto scalare approssimi il logaritmo della probabilità di co-occorrenza.

La funzione obiettivo di GloVe è definita come J = sum_{i,j=1}^{|V|} f(X_{ij}) (v_i^T u_j + b_i + c_j - log X_{ij})^2, dove |V| è la dimensione del vocabolario, b_i e c_j sono termini di bias per la parola i e il contesto j, e f è una funzione di pesatura che riduce l'influenza di co-occorrenze molto rare o molto frequenti. La funzione f è tipicamente definita come f(x) = (x / x_{max})^α se x < x_{max}, e f(x) = 1 altrimenti, con x_{max} tipicamente impostato a 100 e α tipicamente a 0,75. Questa funzione dà peso zero alle co-occorrenze nulle (evitando di doverle processare esplicitamente), peso crescente sublineare alle co-occorrenze rare, e peso massimo alle co-occorrenze moderate, evitando che le co-occorrenze estremamente frequenti dominino l'addestramento.

L'obiettivo di GloVe è formulato come un problema di regressione pesata sui logaritmi delle co-occorrenze. Questa scelta del logaritmo è motivata dall'osservazione empirica che le relazioni semantiche sembrano manifestarsi come differenze nei logaritmi delle probabilità di co-occorrenza. Il logaritmo trasforma inoltre i prodotti in somme, facilitando l'algebra e rendendo l'obiettivo più trattabile.

La derivazione teorica di GloVe parte dal tentativo di codificare i rapporti di probabilità di co-occorrenza nella struttura vettoriale. Gli autori postulano una funzione F che mappa la differenza di vettori di parole e un vettore di contesto al rapporto di probabilità: F(v_i - v_j, u_k) = P_{ik} / P_{jk}, dove P_{ik} = P(k | i) = X_{ik} / X_i è la probabilità che la parola k appaia nel contesto della parola i, e X_i = sum_k X_{ik} è il numero totale di parole nel contesto di i. Imponendo che F preservi la struttura lineare dello spazio vettoriale e rispetti simmetrie naturali, si arriva alla forma F(v_i - v_j, u_k) = exp((v_i - v_j)^T u_k) = exp(v_i^T u_k) / exp(v_j^T u_k), da cui v_i^T u_k = log(X_{ik}) - log(X_i). Assorbendo il termine log(X_i) in un bias e aggiungendo un bias anche per il contesto per simmetria, si ottiene la forma v_i^T u_j + b_i + c_j ≈ log(X_{ij}), che è esattamente ciò che GloVe cerca di soddisfare.

L'addestramento di GloVe procede con discesa del gradiente stocastico o con metodi di ottimizzazione più sofisticati come AdaGrad, che adatta il learning rate per ogni parametro basandosi sulla storia dei gradienti. A differenza di Word2Vec, che processa esempi di addestramento estratti dal corpus in modo streaming, GloVe prima calcola la matrice di co-occorrenza completa e poi ottimizza sui suoi elementi non nulli. Questo rende GloVe più efficiente dal punto di vista della memoria per corpora molto grandi, poiché la matrice di co-occorrenza è tipicamente sparsa (la maggior parte delle coppie di parole non co-occorre mai) e può essere rappresentata in modo compatto.

Gli embeddings finali in GloVe possono essere ottenuti in modi diversi. Una possibilità è usare solo i vettori v_i, trattando u_j puramente come parametri ausiliari. Un'altra è usare solo u_j. Una terza, spesso usata in pratica, è calcolare la media (v_i + u_i) / 2, sfruttando la simmetria approssimativa tra parole e contesti. L'averaging tende a produrre embeddings leggermente più robusti.

GloVe può essere interpretato come una forma di fattorizzazione di matrici: cerca di fattorizzare la matrice log(X) + costanti come prodotto di due matrici di embedding V e U^T. Questa interpretazione connette GloVe a metodi classici di riduzione dimensionale come la Latent Semantic Analysis, che usa la SVD per fattorizzare matrici di co-occorrenza o di rilevanza termine-documento. Tuttavia, GloVe differisce da LSA nell'uso di una funzione di pesatura specifica, nell'aggiunta di termini di bias, e nell'ottimizzazione diretta piuttosto che nell'uso di decomposizioni algebriche.

Gli autori di GloVe hanno condotto esperimenti comparativi dettagliati che mostrano che GloVe produce embeddings di qualità comparabile o superiore a Word2Vec su vari benchmark, con tempi di addestramento spesso più favorevoli per corpora molto grandi. Un vantaggio concettuale di GloVe è la sua trasparenza: l'obiettivo è esplicito e basato su statistiche interpretabili, mentre Word2Vec ottimizza una log-verosimiglianza che è più difficile da collegare direttamente alle statistiche di co-occorrenza.

Tuttavia, analisi successive hanno mostrato che Word2Vec con negative sampling può essere anch'esso interpretato come una fattorizzazione implicita di una matrice di PMI shiftata. Levy e Goldberg hanno dimostrato che ottimizzare l'obiettivo di Skip-gram con negative sampling è equivalente a fattorizzare una matrice i cui elementi sono PMI(w, c) - log(k), dove k è il numero di campioni negativi. Questo suggerisce che Word2Vec e GloVe, nonostante le differenze algoritmiche, condividono fondamenti matematici simili.

Le prestazioni relative di Word2Vec e GloVe dipendono da molti fattori: dimensione e tipo di corpus, scelta di iperparametri, task di valutazione. In generale, entrambi producono embeddings di alta qualità, e la scelta tra i due è spesso dettata da considerazioni pratiche di implementazione e efficienza piuttosto che da differenze qualitative drammatiche. Molti praticanti preferiscono Word2Vec per la sua semplicità concettuale e la disponibilità di implementazioni altamente ottimizzate, mentre GloVe attrae per la sua fondazione su statistiche globali esplicite.

FastText: Embeddings Compositivi a Livello di Subword

FastText, sviluppato da Piotr Bojanowski, Edouard Grave, Armand Joulin e Tomas Mikolov presso Facebook AI Research e pubblicato nel 2017, estende Word2Vec incorporando informazioni morfologiche a livello di carattere. L'idea fondamentale è che le parole non siano unità atomiche ma siano composte da parti significative più piccole, i morfemi, e che la rappresentazione di una parola dovrebbe riflettere questa composizione interna.

FastText rappresenta ogni parola come un insieme di n-grammi di caratteri, oltre alla parola completa. Per esempio, la parola "cane" con n-grammi di lunghezza 3-6 potrebbe essere rappresentata dai trigrammi "<ca", "can", "ane", "ne>", dai 4-grammi "<can", "cane", "ane>", e così via, dove i simboli < e > denotano inizio e fine parola. Ogni n-gramma ha associato un vettore di embedding, e la rappresentazione della parola completa è la somma dei vettori dei suoi n-grammi.

Formalmente, sia G_w l'insieme degli n-grammi della parola w, inclusa la parola stessa. Ogni n-gramma g ∈ G_w ha un vettore z_g ∈ R^d. Il vettore della parola w è definito come v_w = sum_{g ∈ G_w} z_g. Durante l'addestramento, FastText ottimizza una funzione obiettivo simile a quella di Skip-gram, ma invece di aggiornare direttamente v_w, aggiorna i vettori z_g dei n-grammi componenti.

Questa architettura compositiva conferisce a FastText diversi vantaggi significativi. Primo, può generare rappresentazioni per parole fuori vocabolario, ovvero parole non viste durante l'addestramento, purché siano composte da n-grammi conosciuti. Questo è particolarmente utile per lingue morfologicamente ricche dove la derivazione e la flessione producono un vasto numero di forme di parole. Secondo, condivide informazioni morfologica tra parole correlate: "cane", "cani", "canile" condividono n-grammi e quindi hanno rappresentazioni parzialmente sovrapposte che riflettono la loro parentela morfologica. Terzo, è più robusto per parole rare, poiché anche se una parola appare poche volte nel corpus, i suoi n-grammi potrebbero apparire in molte altre parole, accumulando informazioni.

Il numero di n-grammi cresce rapidamente con la lunghezza delle parole e il range di n, portando a un numero totale di n-grammi potenzialmente enorme. Per mantenere la trattabilità computazionale, FastText usa una tecnica di hashing: invece di mantenere un vocabolario esplicito di tutti gli n-grammi possibili, applica una funzione hash agli n-grammi per mapparli a un numero fisso di bucket (tipicamente da uno a due milioni). Molteplici n-grammi possono essere mappati allo stesso bucket (collisioni), ma in pratica questo non degrada significativamente la qualità se il numero di bucket è sufficientemente grande.

L'addestramento di FastText è simile a Word2Vec Skip-gram con negative sampling, ma la complessità computazionale è maggiore perché per ogni parola bisogna sommare i vettori di tutti i suoi n-grammi. Per migliorare l'efficienza, FastText usa varie ottimizzazioni, incluse implementazioni parallelizzate e strategie di campionamento intelligenti.

Gli esperimenti mostrano che FastText produce embeddings di qualità superiore per lingue morfologicamente ricche come il tedesco, il finlandese, il turco, il ceco, dove le parole possono avere molte forme flesse e derivate. Per l'inglese, che ha una morfologia relativamente semplice, i vantaggi sono meno pronunciati ma comunque presenti, specialmente per parole rare e termini tecnici.

Un caso d'uso particolarmente interessante di FastText è il processamento di testo noisy come quello dei social media, dove sono frequenti errori di battitura, abbreviazioni, neologismi. Grazie alla rappresentazione a livello di n-grammi, FastText può generare embeddings ragionevoli anche per forme ortografiche non standard, a patto che condividano n-grammi con parole conosciute.

FastText è stato esteso per supportare anche il multilinguismo: i ricercatori di Facebook AI hanno rilasciato embeddings multilingue addestrati su Wikipedia in oltre 150 lingue. Questi embeddings possono essere allineati tra lingue diverse per supportare applicazioni cross-linguistiche come la traduzione automatica e il recupero di informazioni multilingue.

Una limitazione di FastText è che la composizione tramite somma dei n-grammi è una funzione molto semplice che non cattura interazioni complesse tra le parti. Estensioni recenti

hanno esplorato composizioni più sofisticate usando reti neurali per combinare gli n-grammi, ottenendo miglioramenti ulteriori a costo di complessità maggiore.

Analogie Vettoriali: Relazioni Semantiche come Offset Geometrici

Uno dei risultati più sorprendenti e affascinanti emersi dallo studio degli word embeddings è la scoperta che le relazioni semantiche e sintattiche tra parole si manifestano come regolarità geometriche nello spazio vettoriale. In particolare, le analogie della forma "A sta a B come C sta a D" possono spesso essere risolte attraverso semplici operazioni di aritmetica vettoriale. L'esempio paradigmatico, già menzionato, è l'analogia "re sta a uomo come regina sta a donna", che si traduce nell'equazione vettoriale approssimata v_re - v_uomo + v_donna ≈ v_regina.

Questa proprietà può essere formalizzata nel seguente modo: data un'analogia a:b = c:d, dove stiamo cercando di trovare la parola d che completa l'analogia, l'approccio vettoriale consiste nel calcolare il vettore v* = v_b - v_a + v_c e poi cercare la parola w (escluse a, b, c) il cui vettore v_w massimizza la similarità del coseno con v*: d = argmax_{w ∈ V \ {a,b,c}} cos_sim(v_w, v*). Questo metodo, noto come metodo 3CosAdd (three-term cosine addition), è stato uno dei primi proposti e rimane largamente utilizzato.

Un metodo alternativo, chiamato 3CosMul (three-term cosine multiplication), è stato proposto da Levy e Goldberg per mitigare alcuni problemi del metodo additivo. Il 3CosMul cerca la parola d che massimizza il rapporto delle similarità: d = argmax_{w ∈ V \ {a,b,c}} (cos_sim(v_w, v_b) · cos_sim(v_w, v_c)) / (cos_sim(v_w, v_a) + ε), dove ε è una costante piccola per evitare divisione per zero. Questo metodo tende a produrre risultati leggermente migliori su alcuni benchmark, presumibilmente perché bilancia meglio l'influenza dei tre termini dell'analogia.

Le analogie vettoriali coprono una gamma sorprendentemente ampia di relazioni semantiche e sintattiche. Tra le relazioni semantiche ben catturate troviamo:

- Relazioni di genere: uomo:donna = re:regina = attore:attrice = fratello:sorella
- Relazioni geografiche: Parigi:Francia = Roma:Italia = Tokyo:Giappone = Berlino:Germania
- Relazioni comparativo-superlativo: buono:migliore = grande:più_grande = cattivo:peggiore
- Relazioni singolare-plurale: cane:cani = gatto:gatti = casa:case = libro:libri
- Relazioni verbo-tempo: camminare:camminato = correre:corso = parlare:parlato

Tra le relazioni sintattiche catturate:

- Aggettivo-avverbio: veloce:velocemente = lento:lentamente = felice:felicemente
- Presente-gerundio: andare:andando = mangiare:mangiando = leggere:leggendo
- Sostantivo-aggettivo: Inghilterra:inglese = Spagna:spagnolo = Francia:francese

La regolarità di queste relazioni suggerisce che lo spazio vettoriale apprende a codificare concetti relazionali astratti come "genere", "pluralità", "nazionalità", "tempo verbale" come

direzioni specifiche nello spazio. Il vettore v_re - v_uomo può essere interpretato come il vettore "femminilità nel contesto della regalità", e applicare questo stesso offset a v_donna dovrebbe produrre un vettore vicino a v_regina.

Tuttavia, è importante non sopravvalutare la generalità e la robustezza delle analogie vettoriali. Studi critici hanno evidenziato che le prestazioni sulle analogie dipendono fortemente dalla selezione delle parole, dalla frequenza nel corpus, e dalla formulazione specifica dell'analogia. Analogie che coinvolgono parole molto frequenti e relazioni salienti tendono a funzionare meglio di analogie con parole rare o relazioni più sottili. Inoltre, gli embeddings possono riflettere bias culturali e stereotipi presenti nei dati di addestramento, portando ad analogie problematiche dal punto di vista etico.

Un esempio famoso di bias è l'analogia uomo:programmatore = donna:casalinga, che riflette stereotipi di genere nei dati. Analogie come queste hanno spinto ricerche sul debiasing degli embeddings, cercando di rimuovere o attenuare bias legati a genere, razza, o altre categorie protette preservando al contempo l'utilità semantica generale delle rappresentazioni.

L'interpretazione geometrica delle analogie è che esistono sottospazi o direzioni nello spazio vettoriale corrispondenti a dimensioni semantiche specifiche. Per esempio, potrebbe esistere una "direzione di genere" lungo la quale parole maschili e femminili correlate sono separate. Proiettando una parola lungo questa direzione, si potrebbe ottenere la sua controparte di genere opposto. Ricerche hanno cercato di identificare e caratterizzare esplicitamente questi sottospazi usando tecniche di analisi delle componenti principali applicate a insiemi di coppie di parole che differiscono per una specifica relazione.

Un framework teorico per comprendere le analogie è stato proposto da Ethayarajh e colleghi, che hanno analizzato sotto quali condizioni la relazione a:b = c:d implica v_b - v_a ≈ v_d - v_c. Hanno mostrato che questa approssimazione vettoriale vale quando la relazione è codificata consistentemente come un offset parallelo nello spazio, e che questo accade più facilmente per relazioni morfosintattiche regolari che per relazioni semantiche più complesse e variabili.

Le analogie vettoriali non sono perfette e presentano limitazioni. Primo, sono sensibili alla scala: parole con vettori di magnitudine molto diversa possono produrre offset che non si trasferiscono bene. Secondo, sono sensibili al rumore: l'aggiunta di un offset può portare in regioni dello spazio non ben popolate, dove i vettori più vicini non corrispondono a parole semanticamente rilevanti. Terzo, alcune relazioni semantiche non sono catturabili come semplici offset: relazioni asimmetriche, relazioni contestualmente dipendenti, o relazioni che coinvolgono più di due argomenti.

Nonostante queste limitazioni, la scoperta delle analogie vettoriali ha avuto un impatto profondo sulla ricerca in NLP e sulla comprensione delle rappresentazioni neurali. Ha dimostrato che gli embeddings non catturano solo similarità semantiche generiche ma codificano strutture relazionali ricche. Ha ispirato lo sviluppo di metodi per il reasoning relazionale, per l'estrazione di relazioni da testo, per la costruzione automatica di ontologie.

Sinonimia e Similarità Semantica: Cluster e Regioni nello Spazio

La sinonimia, ovvero la relazione tra parole con significato molto simile o identico, si manifesta negli spazi vettoriali semantici come vicinanza geometrica. Parole sinonime come "felice" e "contento", "auto" e "macchina", "rapido" e "veloce" tendono ad avere vettori con alta similarità del coseno, collocandosi in regioni vicine dello spazio.

Questa proprietà può essere sfruttata per compiti di espansione lessicale, dove dato un termine di query si cercano sinonimi o parole correlate. L'approccio più semplice consiste nel calcolare la similarità del coseno tra il vettore della parola di query e i vettori di tutte le altre parole nel vocabolario, quindi restituire le k parole con similarità più alta. Formalmente, dati una parola w e il suo vettore v_w, le parole più simili sono: {w_i}*{i=1}^k* = *topk*{w' ∈ V \ {w}} cos_sim(v_w, v_{w'}).

Per rendere questa ricerca efficiente in vocabolari grandi, si utilizzano strutture dati specializzate come alberi KD, locality-sensitive hashing, o librerie di ricerca approssimativa di vicini più prossimi come FAISS o Annoy. Queste strutture permettono di trovare i vicini più prossimi in tempo sublineare rispetto alla dimensione del vocabolario, rendendo pratiche applicazioni su larga scala.

Gli esperimenti mostrano che le parole restituite dalla ricerca per similarità includono tipicamente sinonimi veri e propri, ma anche parole correlate semanticamente in modi diversi: co-iponimi (parole che condividono un iperonimo comune, come "cane" e "gatto" che sono entrambi animali), parole che appaiono in contesti simili per ragioni topiche, parole che condividono proprietà o associazioni. Questa ampiezza riflette il fatto che la similarità distribuzionale cattura co-occorrenza contestuale, che correla con ma non è identica alla sinonimia in senso stretto.

Per distinguere tra diversi tipi di relazioni semantiche, sono stati sviluppati metodi che vanno oltre la semplice similarità del coseno. Un approccio consiste nell'usare classificatori supervisionati addestrati su coppie di parole annotate con il tipo di relazione (sinonimia, antinomia, iperonimia, ecc.), usando come feature i vettori delle parole o combinazioni dei vettori. Un altro approccio sfrutta pattern sintattici: parole che appaiono in costruzioni sintattiche parallele sono più probabilmente sinonimi.

La struttura dei cluster semantici nello spazio vettoriale riflette la tassonomia lessicale della lingua. Parole che appartengono alla stessa categoria semantica tendono a formare cluster: nomi di animali, nomi di paesi, verbi di movimento, aggettivi di colore si raggruppano in regioni relativamente coese dello spazio. Questa organizzazione emerge automaticamente dall'addestramento basato sulla distribuzione contestuale, senza supervisione esplicita sulle categorie.

L'analisi dei cluster può rivelare gerarchie semantiche. Per esempio, all'interno del cluster degli animali, possiamo trovare sub-cluster di mammiferi, uccelli, pesci, ciascuno con ulteriori suddivisioni. Questa struttura gerarchica può essere estratta usando tecniche di clustering gerarchico applicate agli embeddings, producendo dendrogrammi che assomigliano a ontologie lessicali costruite manualmente.

La densità locale nello spazio vettoriale varia: alcune regioni sono densamente popolate con molte parole vicine, mentre altre sono più sparse. Le regioni dense tipicamente corrispondono a campi semantici ricchi e produttivi nella lingua, dove esistono molte distinzioni lessicali fini. Le regioni sparse possono corrispondere a concetti periferici o altamente specializzati.

La visualizzazione degli embeddings è una sfida a causa della loro alta dimensionalità. Tecniche di riduzione dimensionale come t-SNE (t-distributed stochastic neighbor embedding) o UMAP (Uniform Manifold Approximation and Projection) possono proiettare gli embeddings da uno spazio di centinaia di dimensioni a uno spazio 2D o 3D visualizzabile. Queste proiezioni preservano approssimativamente le strutture di vicinanza locale, permettendo di osservare visivamente i cluster semantici. Tuttavia, è importante ricordare che le proiezioni introducono distorsioni e la vera struttura geometrica esiste nello spazio ad alta dimensionalità.

Gli studi di visualizzazione hanno rivelato pattern interessanti: parole astratte tendono a occupare regioni diverse da parole concrete; parole con alta valenza emotiva si raggruppano separatamente da parole neutre; termini tecnici di domini specifici formano cluster distinti. Queste osservazioni suggeriscono che gli embeddings catturano non solo relazioni semantiche denotative ma anche connotazioni, registri stilistici, e affiliazioni di dominio.

Antinomia e Relazioni di Opposizione: Geometria della Contraddizione

L'antinomia, la relazione tra parole con significato opposto, presenta una sfida particolare per gli spazi vettoriali semantici basati sulla distribuzione contestuale. Antonimi come "caldo" e "freddo", "alto" e "basso", "amare" e "odiare" tendono a apparire in contesti sintattici e topici molto simili, descrivendo entrambi estremità di una stessa dimensione semantica. Di conseguenza, gli embeddings addestrati puramente sulla base della co-occorrenza contestuale spesso assegnano alta similarità agli antonimi, collocandoli vicini nello spazio.

Questo fenomeno, inizialmente sorprendente, riflette una proprietà genuina del linguaggio: gli antonimi sono semanticamente correlati, anche se in opposizione. Condividono lo stesso dominio di applicazione e spesso appaiono in contrapposizioni esplicite ("né caldo né freddo", "sia alto che basso") che rafforzano la loro co-occorrenza. Gli antonimi sono parole che differiscono su una singola dimensione semantica saliente mantenendo le altre dimensioni costanti.

Per catturare esplicitamente l'antinomia, sono stati sviluppati approcci che vanno oltre la pura distribuzione contestuale. Un metodo consiste nell'usare pattern lessico-sintattici che segnalano opposizione, come "X ma non Y", "da X a Y", "sia X che Y". Parole che appaiono frequentemente in questi pattern hanno alta probabilità di essere antonimi. Integrando questa informazione nell'addestramento degli embeddings, si possono apprendere rappresentazioni dove antonimi sono distinguibili da sinonimi.

Mohammad e colleghi hanno proposto un approccio che usa dizionari di antonimi e sinonimi per vincolare l'addestramento: l'obiettivo viene modificato per penalizzare alta similarità tra

antonimi noti e ricompensare alta similarità tra sinonimi noti. Questo porta a embeddings dove antonimi sono collocati in regioni opposte lungo certe direzioni, mentre sinonimi rimangono vicini.

Un'interpretazione geometrica dell'antinomia è che gli antonimi sono separati lungo una specifica direzione corrispondente alla dimensione semantica su cui si oppongono, ma sono vicini in tutte le altre dimensioni. Per esempio, "caldo" e "freddo" potrebbero differire lungo la direzione della temperatura, ma essere simili nelle dimensioni relative a materialità, fisicità, e applicabilità a oggetti e ambienti. Questa struttura può essere modellata usando spazi vettoriali strutturati dove dimensioni specifiche codificano proprietà specifiche.

La rilevazione automatica di antonimi dagli embeddings è più difficile della rilevazione di sinonimi. Approcci basati su machine learning supervisionato addestrano classificatori binari che distinguono coppie di sinonimi da coppie di antonimi usando feature derivate dai vettori, come la differenza dei vettori, il prodotto elemento per elemento, la concatenazione. Questi classificatori possono raggiungere prestazioni ragionevoli ma richiedono dati di addestramento annotati.

Un'osservazione interessante è che mentre gli antonimi gradabili (che ammettono gradi intermedi, come "caldo-tiepido-freddo") tendono ad avere embeddings relativamente simili, gli antonimi complementari (che si escludono mutuamente senza gradi intermedi, come "vivo-morto", "sposato-celibe") possono essere più separati nello spazio, riflettendo la loro incompatibilità semantica più netta.

La struttura degli antonimi può essere sfruttata per applicazioni come il sentiment analysis. In questo contesto, la polarità semantica (positiva vs negativa) è cruciale. Parole con polarità opposta come "felice" e "triste", "successo" e "fallimento" dovrebbero essere distinguibili negli embeddings per permettere classificazioni accurate del sentimento. Embeddings specifici per il sentiment possono essere addestrati incorporando informazioni di polarità, migliorando le prestazioni su task affettivi.

Relazioni di Iperonimia e Iponomia: Gerarchie Tassonomiche nello Spazio

Le relazioni di iperonimia (is-a) costituiscono la spina dorsale delle tassonomie lessicali e delle ontologie. Un iperonimo è una parola di significato più generale che sussume un iponimo più specifico: "animale" è iperonimo di "cane", "veicolo" è iperonimo di "automobile", "frutto" è iperonimo di "mela". Queste relazioni gerarchiche sono fondamentali per il ragionamento semantico e la categorizzazione.

Catturare l'iperonimia negli spazi vettoriali euclidei standard non è semplice, poiché questi spazi trattano tutte le direzioni simmetricamente. La relazione di iperonimia è intrinsecamente asimmetrica e transitiva: se A è iponimo di B e B è iponimo di C, allora A è iponimo di C. Gli embeddings standard non codificano esplicitamente questa struttura.

Ricerche hanno mostrato che informazioni sull'iperonimia sono parzialmente presenti negli embeddings standard in forma implicita. Per esempio, gli iponimi tendono ad avere vettori di norma maggiore dei loro iperonimi, riflettendo il fatto che i termini più specifici possono

essere associati a contesti più ristretti e distintivi. Inoltre, la differenza dei vettori v_{iponimo} - v_{iperonimo} può catturare la direzione di specificazione, anche se questa regolarità è meno consistente delle analogie vettoriali.

Per migliorare la cattura dell'iperonimia, sono stati proposti embeddings specializzati che usano geometrie non euclidee. I Poincaré embeddings, introdotti da Nickel e Kiela, rappresentano le parole come punti in uno spazio iperbolico, specificamente nel modello del disco di Poincaré. La geometria iperbolica ha la proprietà che il volume di una sfera cresce esponenzialmente con il raggio, permettendo di rappresentare naturalmente strutture gerarchiche ad albero con distorsione minimale.

Nel disco di Poincaré, la distanza iperbolica tra due punti u e v è definita come d_H(u, v) = arccosh(1 + 2 ||u - v||^2 / ((1 - ||u||^2)(1 - ||v||^2))). Questa distanza cresce rapidamente quando i punti si avvicinano al bordo del disco, permettendo di codificare grandi gerarchie in dimensioni relativamente basse. I nodi vicini alla radice di una gerarchia sono collocati vicino al centro del disco, mentre i nodi più specifici e foglie sono collocati più vicino al bordo.

Gli esperimenti con Poincaré embeddings su tassonomie come WordNet hanno mostrato che possono catturare relazioni gerarchiche molto meglio degli embeddings euclidei standard, soprattutto in dimensioni basse. Un embedding 5-dimensionale iperbolico può catturare una gerarchia che richiederebbe centinaia di dimensioni in uno spazio euclideo.

Un approccio alternativo per catturare l'iperonimia negli embeddings euclidei è l'uso di order embeddings, proposti da Vendrov e colleghi. In questo metodo, la relazione di iperonimia è codificata come un ordine parziale sui vettori: x è iponimo di y se e solo se x_i ≤ y_i per ogni dimensione i. Questa rappresentazione sfrutta l'ordine componente per componente per codificare la gerarchia.

Metodi supervisionati per l'estrazione di iperonimia dagli embeddings addestrano classificatori che decidono se una coppia di parole è in relazione di iperonimia. Feature utilizzate includono la concatenazione dei vettori, la differenza, pattern di co-occorrenza in costruzioni sintattiche specifiche come "X such as Y" o "Y is a type of X", e feature basate sulla distribuzione. Questi metodi possono raggiungere accuratezza elevata ma richiedono dati di addestramento annotati.

L'integrazione esplicita di conoscenza tassonomica durante l'addestramento degli embeddings è un'area di ricerca attiva. Approcci retrofitting prendono embeddings pre-addestrati e li raffinano usando vincoli derivati da risorse lessicali come WordNet o Freebase. I vettori vengono aggiustati per avvicinare parole che sono sinonimi e per codificare relazioni di iperonimia secondo pattern geometrici desiderati, preservando al contempo l'informazione distribuzionale originale.

Composizionalità Semantica: Dalla Parola alla Frase

Uno dei principi fondamentali della semantica linguistica è la composizionalità, formulata dal filosofo Gottlob Frege: il significato di un'espressione complessa è funzione del significato delle sue parti costituenti e della loro modalità di combinazione. Negli spazi vettoriali

semantici, questa proprietà si traduce nella sfida di costruire rappresentazioni vettoriali per frasi, proposizioni e documenti a partire dalle rappresentazioni delle singole parole.

L'approccio compositivo più semplice è la somma vettoriale: data una frase composta da parole w_1, w_2, ..., w_n con vettori v_{w_1}, v_{w_2}, ..., v_{w_n}, la rappresentazione della frase è v_frase = sum_{i=1}^n v_{w_i}. Questo metodo, noto come bag-of-words vettoriale, tratta la frase come un insieme non ordinato di parole e produce una rappresentazione che cattura il contenuto lessicale complessivo ma ignora l'ordine e la struttura sintattica.

Nonostante la sua semplicità, la somma vettoriale funziona sorprendentemente bene per molti task, specialmente quando combinata con pesature come il TF-IDF (term frequency-inverse document frequency) che riduce l'influenza di parole molto frequenti e generiche. Una variante è la media vettoriale: v_frase = (1/n) sum_{i=1}^n v_{w_i}, che normalizza rispetto alla lunghezza della frase e può essere più stabile per frasi di lunghezza variabile.

Tuttavia, la somma o media vettoriale presenta limitazioni evidenti. Non distingue tra "il gatto mangia il topo" e "il topo mangia il gatto", producendo la stessa rappresentazione per entrambe nonostante i significati opposti. Non cattura la negazione: "mi piace" e "non mi piace" hanno rappresentazioni molto simili se si usa semplicemente la somma. Non modella modificazioni complesse: l'effetto di un aggettivo su un sostantivo non è catturato adeguatamente dalla somma.

Per affrontare queste limitazioni, sono stati sviluppati metodi compositivi più sofisticati che incorporano informazioni strutturali. Le reti neurali ricorrenti, abbreviate RNN, processano le parole di una frase sequenzialmente, mantenendo uno stato nascosto che viene aggiornato a ogni passo. Dato un vettore di parola v_t e uno stato nascosto precedente h_{t-1}, lo stato nascosto viene aggiornato come h_t = f(W_h h_{t-1} + W_x v_t + b), dove W_h e W_x sono matrici di pesi, b è un vettore di bias, e f è una funzione di attivazione non lineare come la tangente iperbolica. Lo stato nascosto finale h_n cattura una rappresentazione compositiva della frase che tiene conto dell'ordine delle parole.

Le RNN bidirezionali processano la frase in entrambe le direzioni, forward e backward, producendo due sequenze di stati nascosti che vengono poi combinate (tipicamente tramite concatenazione) per ottenere una rappresentazione più ricca che cattura contesto sia precedente che successivo per ogni parola.

Le Long Short-Term Memory networks, abbreviate LSTM, sono una variante delle RNN progettata per mitigare il problema del vanishing gradient che affligge le RNN standard, rendendo difficile l'apprendimento di dipendenze a lungo raggio. Le LSTM introducono celle di memoria e gate (dimenticanza, input, output) che regolano il flusso di informazione, permettendo di mantenere informazioni rilevanti per molti passi temporali. Le equazioni che governano una LSTM sono:

f_t = σ(W_f [h_{t-1}, v_t] + b_f) (gate di dimenticanza) i_t = σ(W_i [h_{t-1}, v_t] + b_i) (gate di input) c̃_t = tanh(W_c [h_{t-1}, v_t] + b_c) (candidato di cella) c_t = f_t ⊙ c_{t-1} + i_t ⊙ c̃_t (stato di cella) o_t = σ(W_o [h_{t-1}, v_t] + b_o) (gate di output) h_t = o_t ⊙ tanh(c_t) (stato nascosto)

dove σ è la funzione sigmoide, ⊙ denota moltiplicazione elemento per elemento, e [h_{t-1}, v_t] indica concatenazione.

Le Gated Recurrent Units, abbreviate GRU, sono una semplificazione delle LSTM con architettura leggermente più semplice ma prestazioni comparabili, usando solo due gate invece di tre e combinando lo stato di cella con lo stato nascosto.

Le reti ricorrenti possono essere addestrate end-to-end per task specifici come sentiment analysis, classificazione di testo, o question answering. Durante l'addestramento, i gradienti della loss rispetto alle rappresentazioni delle frasi si propagano indietro attraverso la rete, aggiornando non solo i pesi della rete ricorrente ma potenzialmente anche gli embeddings delle parole input, permettendo di raffinare le rappresentazioni lessicali in modo task-specific.

Un approccio alternativo alla composizione sequenziale è l'uso di reti convoluzionali, abbreviate CNN, originariamente sviluppate per la visione artificiale ma adattate con successo al processamento del linguaggio. Una CNN applica filtri convoluzionali a finestre di parole consecutive nella frase, estraendo feature locali che catturano pattern n-gramma. Concatenando o pooling le feature estratte da filtri diversi, si ottiene una rappresentazione compositiva della frase.

Le Recursive Neural Networks, da non confondere con le Recurrent Networks, sfruttano la struttura sintattica gerarchica delle frasi. Data una parse tree della frase, una Recursive NN computa rappresentazioni per ogni nodo interno combinando ricorsivamente le rappresentazioni dei figli usando una rete neurale. Per esempio, per un nodo con figli sinistro e destro con rappresentazioni v_L e v_R, la rappresentazione del genitore è v_P = f(W [v_L, v_R] + b). Questo approccio cattura esplicitamente la struttura sintattica compositiva ma richiede parsing sintattico, che può essere costoso o soggetto a errori.

Matrix-Vector Recursive Neural Networks, proposti da Socher e colleghi, estendono le Recursive NN rappresentando ogni parola non solo come un vettore ma come una matrice che codifica come la parola modifica altre parole. La composizione combina le matrici e i vettori, permettendo di catturare modificazioni complesse. Per esempio, un aggettivo ha una matrice che trasforma il sostantivo che modifica.

Un'osservazione importante è che diversi metodi compositivi catturano aspetti diversi del significato frasale. I metodi bag-of-words catturano il contenuto lessicale e topico. Le RNN catturano dipendenze sequenziali e ordine. Le Recursive NN catturano struttura gerarchica. In pratica, la scelta del metodo dipende dal task: per classificazione di topic, un bag-of-words pesato può essere sufficiente; per sentiment analysis che richiede sensibilità alla negazione e all'ordine, una RNN o LSTM è preferibile; per task che richiedono comprensione sintattica profonda, una Recursive NN può essere vantaggiosa.

Embeddings Contestualizzati: ELMo, BERT e Trasformatori

Una limitazione fondamentale degli embeddings tradizionali come Word2Vec e GloVe è che assegnano una rappresentazione statica a ogni parola, indipendentemente dal contesto in

cui appare. Come menzionato precedentemente, questo è problematico per parole polisemiche. La parola "banco" ha un unico vettore che mescola i suoi diversi sensi, anche se in contesti specifici solo uno di questi sensi è rilevante.

Gli embeddings contestualizzati risolvono questo problema generando rappresentazioni diverse per la stessa parola in contesti diversi. ELMo (Embeddings from Language Models), introdotto da Peters e colleghi nel 2018, è stato uno dei primi modelli contestualizzati di successo. ELMo usa una rete neurale LSTM bidirezionale profonda addestrata come modello di linguaggio: il compito è predire la parola successiva data la sequenza precedente (direzione forward) e la parola precedente data la sequenza successiva (direzione backward).

L'architettura di ELMo consiste in molteplici strati LSTM bidirezionali impilati. Per una parola w_t in posizione t di una frase, ciascuno strato produce una rappresentazione contestualizzata basata sull'input e sullo stato nascosto dello strato precedente. La rappresentazione finale di ELMo per w_t è una combinazione pesata delle rappresentazioni da tutti gli strati: v_{w_t} = γ sum_{j=0}^L s_j h_{t,j}, dove L è il numero di strati, h_{t,j} è la rappresentazione dallo strato j, s_j sono pesi di combinazione appresi, e γ è un fattore di scala. I pesi s_j permettono al modello di enfatizzare diversi strati per diversi task: gli strati bassi catturano informazioni sintattiche e morfologiche, mentre gli strati alti catturano semantica più astratta.

ELMo ha dimostrato miglioramenti sostanziali su numerosi task di NLP quando usato per inizializzare o aumentare le rappresentazioni in modelli task-specific. La sua capacità di disambiguare sensi diversi della stessa parola basandosi sul contesto si è rivelata particolarmente preziosa.

BERT (Bidirectional Encoder Representations from Transformers), introdotto da Devlin e colleghi nel 2018, ha rappresentato un ulteriore salto qualitativo. BERT si basa sull'architettura Transformer, che utilizza meccanismi di self-attention invece di ricorrenza per modellare dipendenze in una sequenza. L'attention permette a ogni parola di "prestare attenzione" direttamente a tutte le altre parole nella frase, catturando dipendenze a lungo raggio in modo più efficiente delle RNN.

Il meccanismo di self-attention calcola, per ogni parola, una combinazione pesata delle rappresentazioni di tutte le parole nella sequenza, dove i pesi di attenzione riflettono la rilevanza contestuale. Formalmente, dato un insieme di vettori di input {v_1, v_2, ..., v_n}, self-attention computa per ogni v_i una rappresentazione di output o_i come:

Queries: Q_i = W_Q v_i Keys: K_j = W_K v_j per j = 1, ..., n Values: V_j = W_V v_j per j = 1, ..., n Attention scores: α_{ij} = softmax_j(Q_i · K_j / sqrt(d_k)) Output: o_i = sum_{j=1}^n α_{ij} V_j

dove W_Q, W_K, W_V sono matrici di pesi appresi e d_k è la dimensionalità dei vettori key. I punteggi di attention α_{ij} misurano quanto la parola i dovrebbe prestare attenzione alla parola j, e sono normalizzati tramite softmax per formare una distribuzione di probabilità.

Il Transformer usa multi-head attention, dove molteplici meccanismi di attention operano in parallelo con parametri diversi, catturando diversi tipi di dipendenze. Gli output delle teste

multiple vengono concatenati e proiettati linearmente. Il Transformer impila molteplici strati di multi-head attention, ciascuno seguito da reti feed-forward e normalizzazione.

BERT usa l'architettura Transformer encoder e viene pre-addestrato su due task non supervisionati su enormi corpora testuali:

1. Masked Language Modeling (MLM): una percentuale casuale delle parole input (tipicamente il 15%) viene mascherata (sostituita con un token speciale [MASK]), e il modello deve predire le parole mascherate basandosi sul contesto bidirezionale. Questo forza il modello a apprendere rappresentazioni che codificano informazioni contestuali profonde.

2. Next Sentence Prediction (NSP): dato una coppia di frasi, il modello deve predire se la seconda segue la prima nel testo originale o è una frase casuale. Questo aiuta il modello a catturare relazioni tra frasi.

Dopo il pre-addestramento, BERT può essere fine-tuned su task specifici aggiungendo strati task-specific sull'output e addestrando end-to-end. BERT ha stabilito nuovi stati dell'arte su numerosi benchmark di NLP, dimostrando la potenza del pre-addestramento su larga scala combinato con fine-tuning task-specific.

Le rappresentazioni contestualizzate di BERT per una parola variano drammaticamente in base al contesto. Per esempio, la parola "banco" in "vado in banco" avrà una rappresentazione molto diversa da "banco" in "banco di sabbia", riflettendo i sensi distinti attivati dai contesti.

Modelli successivi hanno esteso e migliorato BERT. RoBERTa ha ottimizzato l'addestramento di BERT rimuovendo il task NSP, usando batch più grandi e più dati. ALBERT ha ridotto il numero di parametri tramite condivisione di parametri tra strati. ELECTRA ha sostituito MLM con un task discriminativo più efficiente. GPT (Generative Pre-trained Transformer) e le sue versioni successive hanno usato un'architettura decoder-only e un obiettivo di modellazione di linguaggio unidirezionale, dimostrando capacità impressionanti di generazione di testo.

XLNet ha combinato idee di modelli autoregressivi come GPT e modelli bidirezionali come BERT usando permutation language modeling, dove tutte le permutazioni dell'ordine delle parole vengono considerate durante l'addestramento, permettendo di catturare dipendenze bidirezionali senza il bias introdotto dai token [MASK].

Gli embeddings contestualizzati hanno cambiato radicalmente il panorama del NLP. Mentre gli embeddings statici come Word2Vec rimangono utili per applicazioni specifiche dove la semplicità e l'efficienza sono prioritarie, gli embeddings contestualizzati rappresentano lo stato dell'arte per la maggior parte dei task che richiedono comprensione linguistica profonda. La loro capacità di disambiguare sensi, catturare sfumature contestuali, e modellare dipendenze complesse li rende strumenti indispensabili per sistemi di NLP moderni.

Analisi della Struttura Geometrica: Manifold e Topologia

Gli spazi vettoriali semantici non sono distribuzioni uniformi di punti nello spazio euclideo, ma esibiscono strutture geometriche complesse che riflettono l'organizzazione del significato linguistico. Comprendere queste strutture richiede strumenti dalla geometria differenziale, dalla topologia, e dall'analisi di manifolds.

Un manifold è uno spazio che localmente assomiglia allo spazio euclideo ma può avere una struttura globale più complessa. L'idea è che, anche se gli embeddings vivono formalmente in uno spazio euclideo ad alta dimensionalità come R^300, i vettori effettivi delle parole potrebbero giacere approssimativamente su un manifold di dimensione intrinseca molto inferiore immerso in questo spazio ambiente. Questa ipotesi del manifold è supportata dall'osservazione che le parole non sono distribuite uniformemente ma formano strutture regolari e correlate.

La dimensione intrinseca di un manifold è il numero minimo di parametri necessari per descrivere localmente il manifold. Se gli embeddings giacciono su un manifold di dimensione intrinseca d_intrinsic molto inferiore alla dimensione ambiente d_ambient, questo suggerisce che il significato linguistico è governato da un numero relativamente piccolo di gradi di libertà, anche se la rappresentazione esplicita usa molte più dimensioni.

Tecniche di stima della dimensione intrinseca possono essere applicate agli embeddings per quantificare questa proprietà. Metodi basati su analisi delle componenti principali esaminano lo spettro dei valori singolari della matrice di embeddings: se i valori singolari decrescono rapidamente, con pochi valori grandi e molti valori piccoli, questo indica bassa dimensione intrinseca. Altri metodi, come l'estimatore di dimensione locale basato su vicini più prossimi, calcolano la dimensione intrinseca in diverse regioni dello spazio, rivelando se la dimensione varia localmente.

Gli studi empirici suggeriscono che la dimensione intrinseca degli spazi di embeddings è sostanzialmente inferiore alla dimensione ambiente, tipicamente dell'ordine di decine piuttosto che centinaia. Questo supporta l'idea che, nonostante l'uso di spazi ad alta dimensionalità per flessibilità e capacità rappresentativa, l'informazione semantica essenziale è codificata in un numero relativamente compatto di direzioni principali.

La curvatura del manifold è un'altra proprietà geometrica rilevante. In uno spazio euclideo piatto, la somma degli angoli interni di un triangolo è sempre 180 gradi e le linee parallele rimangono parallele. In spazi curvi, queste proprietà non valgono. La curvatura può essere positiva (come su una sfera), negativa (come in uno spazio iperbolico), o zero (euclideo). La struttura gerarchica delle tassonomie lessicali, con biforcazioni multiple e crescita esponenziale del numero di nodi ai livelli inferiori, suggerisce che una geometria iperbolica (curvatura negativa) potrebbe essere più naturale per rappresentare questi dati, come dimostrato dai Poincaré embeddings discussi precedentemente.

L'analisi topologica degli spazi di embeddings studia proprietà che sono invarianti sotto deformazioni continue, come il numero di componenti connesse, la presenza di buchi o cavità di diverse dimensioni. La topological data analysis, abbreviata TDA, fornisce strumenti

come la persistent homology per caratterizzare queste proprietà. Applicata agli embeddings, la TDA può rivelare se esistono sottospazi disconnessi corrispondenti a campi semantici distinti, o se esistono strutture cicliche che potrebbero riflettare relazioni semantiche complesse.

La persistenza topologica costruisce una sequenza di spazi topologici a diverse scale, variando un parametro di soglia (tipicamente la distanza), e traccia quando feature topologiche (componenti connesse, cicli, cavità) appaiono e scompaiono. Feature che persistono attraverso un ampio range di scale sono considerate significative, mentre feature che appaiono e scompaiono rapidamente sono considerate rumore. I persistence diagrams e persistence barcodes visualizzano queste informazioni, permettendo di identificare strutture topologiche rilevanti negli embeddings.

Studi usando TDA hanno rivelato che gli spazi di embeddings esibiscono strutture topologiche non triviali, inclusi cicli persistenti che potrebbero corrispondere a campi semantici circolari (per esempio, giorni della settimana, mesi dell'anno) o a relazioni di opposizione che formano strutture bipolari. Cavità di dimensione superiore potrebbero riflettere organizzazioni gerarchiche o categoriali complesse.

La geodesia su manifolds definisce il concetto di "cammino più breve" tra due punti. In uno spazio euclideo, la geodetica è una linea retta. Su un manifold curvo, le geodetiche seguono la curvatura del manifold. Nel contesto degli embeddings, le geodetiche potrebbero rappresentare "percorsi semantici" naturali tra concetti, passando attraverso concetti intermedi in modo fluido. Per esempio, la geodetica da "cane" a "pesce" potrebbe passare attraverso "animale domestico", "animale", "creatura acquatica", riflettendo una transizione semantica graduale.

L'analisi di cluster gerarchica può essere interpretata geometricamente come la costruzione di un dendrogramma che riflette la struttura ad albero delle relazioni di similarità. Gli algoritmi di clustering come k-means partizionano lo spazio in regioni di Voronoi, mentre metodi basati su densità come DBSCAN identificano regioni densamente popolate separate da regioni sparse. Applicati agli embeddings, questi metodi rivelano la granularità e la struttura dei campi semantici.

Bias e Stereotipi negli Embeddings: Problemi Etici e Sociali

Una conseguenza inevitabile dell'ipotesi distribuzionale è che gli embeddings assorbono e codificano i bias presenti nei dati di addestramento. Se un corpus testuale riflette stereotipi di genere, pregiudizi razziali, o altre forme di discriminazione sociale, gli embeddings appresi da quel corpus rifletteranno necessariamente questi pattern. Questa proprietà ha generato un'intensa discussione nella comunità di ricerca su questioni di equità, responsabilità e impatto sociale dei sistemi di intelligenza artificiale.

Uno studio seminale di Bolukbasi e colleghi nel 2016 ha dimostrato che gli embeddings standard di Word2Vec addestrati su Google News esibiscono bias di genere sistematici. Analizzando le analogie vettoriali, hanno trovato che "uomo sta a programmatore come

donna sta a casalinga", "padre sta a medico come madre sta a infermiera", rivelando associazioni che riflettono stereotipi occupazionali di genere. Hanno inoltre mostrato che parole neutri rispetto al genere come professioni o aggettivi di personalità hanno componenti sistematiche lungo una "direzione di genere" nello spazio vettoriale.

La direzione di genere può essere identificata calcolando la prima componente principale di un insieme di coppie di parole che differiscono solo per il genere, come (uomo, donna), (lui, lei), (padre, madre), (re, regina). Questa direzione cattura la variazione principale che distingue parole maschili da parole femminili. Proiettando parole presumibilmente neutre come professioni su questa direzione, si può misurare il loro "bias di genere implicito". Bolukbasi e colleghi hanno trovato che "programmatore" è fortemente spostato verso il polo maschile, mentre "casalinga" è spostata verso il polo femminile, coerentemente con gli stereotipi sociali.

Questo tipo di bias può avere conseguenze pratiche serie quando gli embeddings vengono usati in applicazioni reali. Sistemi di raccomandazione di lavoro basati su embeddings potrebbero discriminare candidature femminili per posizioni tecniche, o candidature maschili per posizioni di cura. Sistemi di traduzione automatica potrebbero assegnare pronomi di genere basandosi su stereotipi occupazionali piuttosto che su informazioni contestuali. Motori di ricerca potrebbero rankare risultati in modi che rafforzano associazioni problematiche.

Il bias non si limita al genere. Studi hanno documentato bias razziali, con parole associate a nomi afroamericani correlate più fortemente con parole negative rispetto a parole associate a nomi europeo-americani. Bias relativi all'età, alla disabilità, alla religione, all'orientamento sessuale sono stati anch'essi identificati. In molti casi, questi bias riflettono accuratamente i pattern presenti nei dati, che a loro volta riflettono pregiudizi sociali reali.

Il debiasing degli embeddings è diventato un'area di ricerca attiva. Bolukbasi e colleghi hanno proposto un metodo che prima identifica una sottospazio di bias (per esempio, la direzione di genere), poi neutralizza le parole che dovrebbero essere neutre proiettandole ortogonalmente a questo sottospazio, e infine equalizza le coppie di parole che dovrebbero differire solo per l'attributo protetto (come professioni) per garantire che abbiano la stessa distanza dal sottospazio neutro.

Formalmente, data una direzione di bias b (vettore unitario), la componente di un vettore v lungo b è proj_b(v) = (v · b) b, e la componente ortogonale è v_⊥ = v - proj_b(v). Il debiasing neutralizza parole sostituendo v con v_⊥ / ||v_⊥||, rimuovendo la componente lungo la direzione di bias e rinormalizzando.

Tuttavia, approcci di debiasing sollevano questioni complesse. Primo, quale definizione di fairness adottare? Neutralizzare completamente le differenze di genere può eliminare informazioni legittime, per esempio in contesti storici o letterari dove il genere è rilevante. Secondo, il debiasing può introdurre nuovi problemi: rimuovere un tipo di bias può amplificarne altri o degradare la performance complessiva. Terzo, il debiasing post-hoc tratta i sintomi ma non le cause: il bias origina dai dati di addestramento che riflettono la società.

Un approccio alternativo è l'adversarial debiasing durante l'addestramento, dove un discriminatore cerca di predire attributi protetti dalle rappresentazioni, e l'encoder cerca di

produrre rappresentazioni che ingannano il discriminatore, risultando in embeddings che non codificano esplicitamente l'attributo protetto. Questo approccio basato su reti adversariali può essere più efficace del debiasing post-hoc ma è anche più complesso da implementare.

La questione del bias solleva domande fondamentali sulla neutralità e l'oggettività del linguaggio e delle rappresentazioni computazionali. Il linguaggio non è uno specchio neutro della realtà ma è intrinsecamente intrecciato con valori sociali, relazioni di potere, e prospettive culturali. Pretendere che gli embeddings siano "neutrali" può essere illusorio e potenzialmente dannoso, mascherando bias sotto un'apparenza di oggettività scientifica.

Un approccio più sfumato riconosce che bias diversi possono essere appropriati per applicazioni diverse. Per un sistema di analisi storica di testi del diciannovesimo secolo, catturare accuratamente i bias dell'epoca può essere desiderabile per comprendere le attitudini sociali storiche. Per un sistema di raccomandazione di assunzioni moderno, eliminare bias discriminatori è un imperativo etico. La chiave è la trasparenza, la consapevolezza, e la possibilità di audit: gli sviluppatori e gli utenti devono essere consapevoli dei bias presenti nei modelli e delle loro possibili conseguenze.

Recenti ricerche hanno anche esplorato il concetto di fairness contestuale: piuttosto che rimuovere globalmente tutti i segnali relativi ad attributi protetti, si cerca di rimuoverli solo quando sono irrilevanti per il compito, preservandoli quando sono informativi in modo legittimo. Questo richiede definizioni più sofisticate di fairness e metodi più fini di intervento sulle rappresentazioni.

La documentazione dei dataset e dei modelli, inclusi i possibili bias, è diventata una pratica raccomandata. Iniziative come datasheets for datasets e model cards forniscono template per documentare le origini dei dati, le procedure di raccolta, i bias noti, le limitazioni, e gli usi raccomandati e sconsigliati. Questa trasparenza permette agli utenti di fare scelte informate e di sviluppare strategie di mitigazione appropriate per le loro applicazioni specifiche.

Applicazioni degli Word Embeddings: Dal Recupero Informazioni alla Traduzione

Gli word embeddings hanno trovato applicazione in una vasta gamma di task di processamento del linguaggio naturale, diventando componenti fondamentali di sistemi moderni. Esploreremo alcune delle applicazioni più significative, evidenziando come la geometria del significato catturata dagli embeddings venga sfruttata in contesti pratici.

Nel recupero di informazioni e nella ricerca semantica, gli embeddings permettono di andare oltre il semplice matching di parole chiave. Una query come "veicolo a quattro ruote" può recuperare documenti che contengono parole come "automobile", "macchina", "auto" anche se queste parole non appaiono nella query, sfruttando la vicinanza semantica nello spazio vettoriale. Il processo tipicamente coinvolge la rappresentazione sia della query che dei documenti come vettori (sommando o mediando gli embeddings delle parole costituenti, o usando rappresentazioni più sofisticate), poi calcolando la similarità del coseno tra il vettore della query e i vettori dei documenti, rankando i documenti per similarità decrescente.

Sistemi di question answering utilizzano embeddings per matching semantico tra domande e possibili risposte. Data una domanda "Chi ha inventato il telefono?", il sistema può identificare passaggi testuali rilevanti cercando regioni dello spazio vettoriale dove concetti come "inventore", "telefono", "Alexander Graham Bell" sono vicini. Gli embeddings aiutano a gestire variazioni parafrastiche: la domanda può essere formulata in molti modi diversi, e la risposta può apparire in forme testuali variegate.

Nella classificazione di testo, gli embeddings forniscono rappresentazioni dense delle parole input che vengono processate da classificatori (regressione logistica, SVM, reti neurali) per predire categorie. Per esempio, in sentiment analysis, frasi positive come "questo film è fantastico" e "mi è piaciuto molto" hanno rappresentazioni vettoriali simili che facilitano la classificazione corretta. L'uso di embeddings pre-addestrati come Word2Vec o GloVe come inizializzazione, seguiti da fine-tuning task-specific, è una pratica comune che sfrutta la conoscenza lessicale generale appresa da grandi corpora non annotati.

Il named entity recognition, abbreviato NER, identifica e classifica entità nominate in testo (persone, organizzazioni, luoghi). Gli embeddings forniscono feature contestuali ricche che aiutano a disambiguare. Per esempio, "Washington" può riferirsi a George Washington (persona), a Washington D.C. (luogo), o a lo stato di Washington (luogo), e il contesto circostante catturato dagli embeddings aiuta la classificazione corretta.

Nella traduzione automatica neurale, gli embeddings sono componenti cruciali delle architetture encoder-decoder. L'encoder traduce la frase sorgente in una rappresentazione vettoriale continua, e il decoder genera la traduzione target basandosi su questa rappresentazione. I modelli di attenzione, che permettono al decoder di "prestare attenzione" a parti diverse della frase sorgente mentre genera ciascuna parola target, si basano crucialmente su similarità e allineamenti nello spazio vettoriale.

La cross-lingual embedding alignment estende gli embeddings monolingue a contesti multilingue. L'idea è apprendere una trasformazione (tipicamente una matrice ortogonale) che allinea gli spazi di embeddings di due lingue diverse, mappando parole traduttive vicine nello spazio condiviso. Questo permette di trasferire conoscenza lessicale tra lingue e di supportare applicazioni come traduzione senza risorse parallele, recupero di informazioni cross-lingua, e transfer learning multilingue.

Il summarization automatico genera riassunti concisi di documenti lunghi. Approcci basati su embeddings possono identificare frasi salienti calcolando la centralità semantica: frasi i cui vettori sono centrali nella distribuzione delle rappresentazioni di tutte le frasi sono probabilmente riassunti buoni perché catturano temi dominanti. Approcci più sofisticati usano reti neurali che generano riassunti astrattivi parola per parola, guidati da rappresentazioni vettoriali del documento sorgente.

Il dialogue e conversational AI sfruttano embeddings per comprendere intenzioni dell'utente, mantenere coerenza contestuale, e generare risposte appropriate. Gli assistenti virtuali usano embeddings per mappare utterances dell'utente a intenzioni predefinite (per esempio, "trova un ristorante", "controlla il tempo"), gestendo variazioni parafrastiche. Durante la generazione di risposte, embeddings guidano la selezione di parole semanticamente coerenti con il contesto.

Nell'information extraction, che mira a estrarre fatti strutturati da testo non strutturato (per esempio, estrarre relazioni come "Barack Obama è nato a Honolulu"), embeddings aiutano a identificare pattern e relazioni. Vettori di coppie di entità che condividono la stessa relazione tendono a occupare regioni simili dello spazio, facilitando la classificazione di relazioni.

La generazione di linguaggio naturale, abbreviata NLG, produce testo fluente e coerente da rappresentazioni strutturate o da contesti dati. I modelli generativi moderni come GPT usano embeddings contestualizzati e meccanismi di attenzione per predire sequenze di parole che sono sia grammaticalmente corrette che semanticamente coerenti. La geometria dello spazio di embeddings guida le scelte lessicali, favorendo parole che sono vicine nello spazio alle rappresentazioni contestuali.

Gli embeddings hanno anche applicazioni al di fuori del NLP tradizionale. Nella raccomandazione di contenuti, embeddings di parole o di entità possono rappresentare interessi degli utenti e attributi degli item, e la similarità nello spazio vettoriale guida le raccomandazioni. Nell'analisi di reti sociali, embeddings possono rappresentare nodi di un grafo in modo che nodi simili o connessi siano vicini nello spazio, facilitando task come community detection e link prediction.

Limitazioni e Critiche: Cosa gli Embeddings Non Catturano

Nonostante i loro successi, gli word embeddings presentano limitazioni significative che è importante riconoscere per evitare applicazioni inappropriate e per guidare ricerche future.

Una limitazione fondamentale è l'assenza di grounding nel mondo fisico. Gli embeddings sono appresi puramente da testo e quindi catturano solo relazioni distribuzionali tra simboli linguistici. Non hanno accesso diretto a esperienze percettive, sensoriali, o motorie che sono parte integrale della cognizione umana. Il significato di "rosso" negli embeddings deriva da come "rosso" co-occorre con altre parole in testo, non dall'esperienza visiva del colore rosso. Questo solleva la questione se gli embeddings catturano davvero "significato" nel senso pieno, o solo una forma superficiale di correlazione statistica.

Il philosophy of language ha discusso a lungo il problema del grounding: come i simboli acquisiscono significato? La posizione embodied cognition sostiene che il significato è radicato nell'esperienza corporea e nell'interazione con il mondo fisico. Da questa prospettiva, rappresentazioni puramente linguistiche sono inevitabilmente incomplete. Recenti sviluppi in AI multimodale che integrano visione, linguaggio, e eventualmente altre modalità rappresentano un passo verso grounding più ricco, ma rimangono limitati rispetto alla ricchezza dell'esperienza umana.

Gli embeddings statici non catturano la polisemia e la variabilità contestuale del significato. Come discusso, gli embeddings contestualizzati come BERT affrontano questo problema, ma introducono complessità computazionale maggiore e perdono alcune delle proprietà algebriche pulite degli embeddings statici. Inoltre, anche gli embeddings contestualizzati non catturano completamente fenomeni come ironia, metafora, o uso creativo del linguaggio che dipendono pesantemente da conoscenza del mondo e pragmatica.

La composizionalità rimane una sfida. Mentre abbiamo discusso vari metodi per comporre embeddings di parole in rappresentazioni di frasi, nessuno di questi metodi cattura completamente le complessità della composizione semantica studiata in semantica formale. Fenomeni come quantificazione ("tutti i gatti sono animali"), negazione ("non mi piace"), modalità ("potrebbe piovere"), tempi verbali complessi, richiedono trattamenti compositivi sofisticati che vanno oltre semplici combinazioni vettoriali.

Gli embeddings codificano informazioni statistiche ma non logiche. Non rappresentano esplicitamente implicazioni logiche, contraddizioni, o relazioni di implicazione. Se il testo dice "John è celibe" e "John è sposato", gli embeddings non riconoscono la contraddizione logica. Approcci che integrano rappresentazioni neurali con ragionamento simbolico sono un'area di ricerca attiva che cerca di combinare i punti di forza di entrambi i paradigmi.

La mancanza di interpretabilità è un'altra limitazione. Le dimensioni degli embeddings non hanno tipicamente interpretazioni semantiche dirette: non possiamo puntare a una dimensione e dire "questa rappresenta animatezza" o "questa rappresenta concretezza". Alcuni sforzi hanno cercato di rendere gli embeddings più interpretabili usando sparse coding o altri metodi che incoraggiano rappresentazioni sparse, ma questo rimane difficile.

Gli embeddings riflettono i dati di addestramento e quindi sono limitati alla conoscenza presente in quei dati. Se un corpus non contiene informazioni su eventi recenti, tecnologie emergenti, o conoscenza specialistica di un dominio, gli embeddings non cattureranno questi aspetti. Questo rende necessario re-addestramento periodico o adattamento di dominio per mantenere gli embeddings rilevanti e accurati.

La sensibilità a rumore e errori nei dati di addestramento è problematica. Errori di spelling, grammatica, o uso atipico del linguaggio nei corpora possono distorcere gli embeddings. Nei testi di social media, pieni di abbreviazioni, slang, e creatività linguistica, la qualità degli embeddings può degradare.

Gli embeddings trattano parole come unità atomiche (a meno che si usino approcci subword come FastText), ma la morfologia e la struttura interna delle parole portano significato. Parole composte, derivazioni, flessioni hanno strutture sistematiche che gli embeddings potrebbero non catturare completamente.

La questione della causalità versus correlazione è fondamentale. Gli embeddings catturano correlazioni distribuzionali ma non distinguono tra relazioni causali, correlazioni spurie, o confounding. Due parole possono co-occorrere frequentemente per ragioni accidentali o a causa di confounders, e l'embedding non ha modo di distinguere questi casi.

Infine, gli embeddings rappresentano una media o sintesi su molti usi di una parola nel corpus, perdendo informazioni su variabilità, eccezionalità, o usi creativi. L'uso linguistico umano è ricco di innovazione, gioco di parole, ambiguità deliberata, che si perdono nell'aggregazione statistica.

Implicazioni Teoriche: Cosa gli Embeddings Rivelano sul Linguaggio e la Cognizione

Al di là delle applicazioni pratiche, gli word embeddings offrono spunti teorici affascinanti sulla natura del linguaggio e potenzialmente sulla cognizione umana. L'emergere di strutture geometriche regolari che riflettono relazioni semantiche solleva domande profonde.

L'ipotesi distribuzionale, confermata dal successo degli embeddings, suggerisce che molto del significato linguistico è codificabile attraverso pattern di co-occorrenza. Questo è coerente con teorie linguistiche e cognitive che enfatizzano l'uso e il contesto. La semantica d'uso di Wittgenstein, che sostiene che il significato è determinato dall'uso nel linguaggio, risuona con l'ipotesi distribuzionale. Se il significato è veramente una funzione dell'uso, allora analizzare statisticamente l'uso dovrebbe rivelare la struttura del significato, che è esattamente ciò che gli embeddings fanno.

La scoperta di analogie vettoriali suggerisce che le relazioni concettuali potrebbero essere rappresentate nella mente umana in modo geometrico o spaziale. Studi di neuroimaging hanno mostrato che rappresentazioni distribuite di concetti nel cervello esibiscono organizzazioni simili a quelle osservate negli embeddings, con categorie semantiche correlate attivando pattern neurali vicini. Questo non implica che il cervello usi esattamente gli stessi algoritmi di Word2Vec, ma suggerisce principi organizzativi condivisi: rappresentazione distribuita, encoding relazionale, similarità basata su sovrapposizione di pattern.

Gli embeddings supportano l'idea di rappresentazioni distribuite in neuroscienze cognitive, proposta inizialmente dai connessionisti. Invece di rappresentare concetti come simboli discreti in specifiche "neuroni della nonna", i concetti sono distribuiti su molte unità, con ogni unità contribuendo a molteplici concetti. Questa visione è più biologicamente plausibile e robusta a danni, e gli embeddings dimostrano che rappresentazioni distribuite possono catturare strutture semantiche complesse.

La struttura gerarchica e tassonomica che emerge negli embeddings riflette organizzazioni categoriali fondamentali del pensiero umano. La psicologia cognitiva ha documentato estesamente come gli esseri umani organizzano concetti in gerarchie (animali -> mammiferi -> cani -> Labrador), e gli embeddings catturano queste gerarchie in modo non supervisionato. Questo suggerisce che l'organizzazione categoriale non è puramente innata o arbitraria ma emerge naturalmente dall'esperienza linguistica e percettiva.

La questione dell'innatismo versus apprendimento è centrale in linguistica e scienze cognitive. Chomsky ha sostenuto che molto della struttura linguistica è innata, codificata geneticamente in una Grammatica Universale. La prospettiva empirista sostiene che la struttura linguistica è appresa dall'esperienza. Gli embeddings dimostrano che strutture semantiche ricche possono emergere da apprendimento statistico su dati linguistici, supportando posizioni empiriste o costruttiviste. Tuttavia, gli algoritmi stessi (architetture neurali, funzioni obiettivo) incorporano bias induttivi che guidano l'apprendimento, quindi la questione rimane complessa.

Gli embeddings toccano anche la filosofia del significato. La semantica referenziale sostiene che il significato di una parola è il suo referente nel mondo. La semantica mentale sostiene che il significato è una rappresentazione mentale. La semantica d'uso sostiene che il significato è determinato dall'uso nel linguaggio. Gli embeddings si allineano più naturalmente con la semantica d'uso, ma sollevano la questione se una rappresentazione

puramente statistica catturi "significato" in senso pieno o solo una sua approssimazione strumentale.

Il dibattito tra rappresentazioni simboliche versus subsimboliche in AI è illuminato dagli embeddings. Il paradigma simbolico rappresenta conoscenza usando simboli discreti e regole logiche. Il paradigma subsimbolico (connessionista) usa rappresentazioni distribuite e apprendimento statistico. Gli embeddings sono quintessenzialmente subsimbolici, ma catturano relazioni che sembrano simboliche (come analogie). Questo suggerisce che la distinzione potrebbe essere meno netta di quanto pensato, o che i due livelli possono coesistere con le rappresentazioni subsimboliche che implementano o approssimano funzionalità simboliche.

La linguistica computazionale ha tradizionalmente enfatizzato sintassi e grammatica formale. Gli embeddings, emergendo dall'elaborazione statistica di corpus, catturano aspetti semantici senza analisi sintattica esplicita, sfidando l'idea che la sintassi sia primaria. Naturalmente, modelli più recenti integrano informazioni sintattiche e semantiche, riconoscendo che entrambe sono essenziali per la comprensione del linguaggio completa.

Gli embeddings offrono anche spunti metodologici: dimostrano il potere dell'apprendimento non supervisionato su larga scala. Gran parte della conoscenza semantica può essere estratta da testo grezzo senza annotazione umana costosa. Questo ha implicazioni per la ricerca in AI e scienze cognitive: piuttosto che costruire manualmente ontologie o regole, si può apprendere molto dall'analisi statistica di dati naturalistici.

Valutazione degli Embeddings: Metriche e Benchmark

La valutazione della qualità degli word embeddings è cruciale per confrontare metodi diversi, per guidare la scelta di iperparametri, e per comprendere quali aspetti del significato vengono catturati. Le metodologie di valutazione si dividono in due categorie principali: valutazione intrinseca, che misura proprietà degli embeddings direttamente, e valutazione estrinseca, che misura le prestazioni su task applicativi a valle.

La valutazione intrinseca tipicamente si basa su benchmark che testano capacità specifiche degli embeddings senza richiedere addestramento di modelli task-specific complessi. I benchmark di similarità semantica forniscono coppie di parole annotate da giudici umani con punteggi di similarità su scale continue (per esempio, da 0 a 10). Dataset classici includono WordSim-353, che contiene 353 coppie di parole inglesi con giudizi di similarità, SimLex-999 che distingue esplicitamente similarità da relatedness, e RG-65 che è un dataset più piccolo ma frequentemente usato. La valutazione calcola la correlazione di Spearman o Pearson tra le similarità del coseno degli embeddings e i giudizi umani. Correlazioni alte indicano che gli embeddings catturano intuizioni umane di similarità semantica.

Tuttavia, questi benchmark presentano limitazioni. I giudizi umani di similarità sono soggettivi e variabili: diversi annotatori possono avere opinioni diverse, e la nozione stessa di "similarità" è multidimensionale e contestualmente dipendente. Inoltre, dataset piccoli possono non essere rappresentativi della diversità lessicale complessiva, e la maggior parte dei dataset esistenti è per l'inglese, limitando la valutazione cross-linguistica.

I benchmark di analogia testano la capacità di risolvere analogie della forma "a sta a b come c sta a d". Il Google Analogy Test Set contiene migliaia di analogie semantiche (capitali-paesi, genere, comparativi) e sintattiche (plurali, tempi verbali, aggettivi-avverbi). La metrica di valutazione è l'accuracy: la frazione di analogie per cui il metodo 3CosAdd o 3CosMul recupera la risposta corretta come prima scelta (escluse le parole della query). Dataset alternativi includono BATS (Bigger Analogy Test Set) e analogie specifiche per lingue diverse dall'inglese.

Le critiche ai benchmark di analogia evidenziano che spesso testano conoscenza enciclopedica o fattuale (capitali, confini nazionali) piuttosto che relazioni semantiche pure. Inoltre, il successo sulle analogie può dipendere fortemente dalla frequenza delle parole nel corpus di addestramento: analogie con parole rare tendono a fallire non per deficit di comprensione relazionale ma per insufficiente evidenza statistica.

La valutazione su task di categorizzazione testa se embeddings catturano appartenenza categoriale. Dataset come AP (Almuhareb-Poesio) contengono parole etichettate con categorie semantiche, e la valutazione misura quanto bene il clustering degli embeddings corrisponde alle categorie annotate, usando metriche come purezza dei cluster o normalized mutual information.

I benchmark di outlier detection presentano insiemi di parole dove una parola è semanticamente fuori luogo, e il sistema deve identificarla. Per esempio, nell'insieme {mela, banana, arancia, macchina, pera}, "macchina" è l'outlier. La valutazione misura l'accuracy di identificazione dell'outlier basandosi sulla distanza media nello spazio di embeddings.

La valutazione estrinseca misura quanto gli embeddings migliorano le prestazioni su task di NLP a valle. Task comuni includono sentiment analysis (classificare testi come positivi o negativi), named entity recognition (identificare nomi di persone, luoghi, organizzazioni), part-of-speech tagging (etichettare parole con categorie grammaticali), parsing sintattico, question answering, e traduzione automatica. Gli embeddings pre-addestrati vengono usati come inizializzazione o feature input per modelli task-specific, e le prestazioni vengono confrontate con baseline che usano rappresentazioni alternative (one-hot, embeddings casuali, altri metodi di pre-training).

La valutazione estrinseca fornisce evidenza diretta dell'utilità pratica degli embeddings ma è più costosa computazionalmente e metodologicamente complessa. Richiede dataset annotati per i task, modelli task-specific, e procedure di addestramento e valutazione standardizzate. Inoltre, le prestazioni dipendono non solo dalla qualità degli embeddings ma anche dall'architettura del modello task-specific, dagli iperparametri, e dai dati di addestramento.

Un aspetto critico della valutazione è la distinzione tra in-domain e out-of-domain performance. Embeddings addestrati su un corpus di news potrebbero performare bene su task di news classification ma male su task medici o legali. La robustezza cross-domain è desiderabile ma difficile da ottenere senza addestramento su corpora diversificati.

La valutazione dovrebbe anche considerare efficienza computazionale: tempo di addestramento, uso di memoria, velocità di inferenza. Embeddings che ottengono qualità

leggermente superiore ma richiedono ordini di grandezza più risorse potrebbero non essere praticamente preferibili per molte applicazioni.

Recentemente, sono emerse preoccupazioni sulla validità dei benchmark standard. Alcuni dataset sono vecchi e potrebbero contenere errori o bias. Le metriche aggregate possono mascherare debolezze su sottoinsiemi specifici. C'è rischio di overfitting ai benchmark: metodi ottimizzati eccessivamente per performare bene su specifici test set potrebbero non generalizzare a usi reali. Questo ha portato a chiamate per benchmark più diversificati, realistici, e dinamici.

Direzioni Future: Verso Rappresentazioni Semantiche più Ricche

La ricerca sugli spazi vettoriali semantici è in costante evoluzione, con numerose direzioni promettenti che cercano di superare le limitazioni attuali e di espandere le capacità rappresentative.

L'integrazione multimodale è una frontiera particolarmente eccitante. Modelli come CLIP (Contrastive Language-Image Pre-training) apprendono rappresentazioni congiunte di testo e immagini, permettendo di collegare concetti linguistici a esperienze visive. L'architettura di CLIP usa un encoder testuale (Transformer) e un encoder visivo (Vision Transformer o ResNet) addestrati contrastivamente per massimizzare la similarità tra embeddings di immagini e delle loro descrizioni testuali. Questo grounding visivo può migliorare la comprensione di concetti concreti e permettere applicazioni come ricerca di immagini da testo, generazione di didascalie, e question answering visivo.

Estensioni a altre modalità sono in esplorazione: audio per concetti legati a suoni e musica, dati tattili per concetti legati a texture e forze, dati motori per azioni e movimenti. L'idea è che rappresentazioni veramente ricche del significato dovrebbero essere grounded in esperienze multisensoriali, riflettendo come gli esseri umani apprendono concetti attraverso interazione corporea con il mondo.

Gli embeddings dinamici e adattivi che evolvono con il tempo catturano il fatto che il linguaggio cambia: nuove parole emergono, significati si spostano, associazioni evolvono. Modelli diacronici tracciano l'evoluzione semantica nel tempo analizzando corpora da epoche diverse. Per esempio, la parola "gay" ha subito uno spostamento semantico drammatico nel corso del ventesimo secolo. Embeddings temporizzati possono catturare questi cambiamenti e essere utili per analisi storica, rilevamento di tendenze emergenti, e monitoraggio di discorsi sociali.

L'incorporazione esplicita di conoscenza strutturata da knowledge graphs come Wikidata, ConceptNet, o ontologie di dominio è un'area attiva. Questi grafi contengono relazioni esplicite tra entità (is-a, part-of, causes, located-in) che complementano le relazioni distribuzionali catturate da testo. Metodi di knowledge graph embedding rappresentano entità e relazioni come vettori e matrici in modo che relazioni vere abbiano score alti. L'integrazione di questi embeddings con word embeddings può produrre rappresentazioni che combinano conoscenza distribuzionale e strutturata.

Gli embeddings probabilistici o Bayesiani modellano l'incertezza nelle rappresentazioni. Invece di assegnare un singolo vettore deterministic a una parola, si assegna una distribuzione di probabilità (per esempio, una Gaussiana caratterizzata da media e varianza). Questo permette di rappresentare l'incertezza dovuta a dati limitati (parole rare hanno alta varianza) e di fare inferenza probabilistica. Vilnis e McCallum hanno proposto Gaussian embeddings dove similarità è misurata usando divergenze tra distribuzioni.

Le rappresentazioni gerarchiche e composizionali più sofisticate cercano di catturare la struttura interna dei concetti. I embeddings hypernymy-aware usano geometrie o meccanismi specifici per modellare relazioni is-a. I tensor embeddings usano tensori (array multidimensionali) invece di vettori per rappresentare parole, permettendo di catturare interazioni più complesse. Le capsule networks, proposte da Hinton, rappresentano concetti come vettori di parametri che codificano proprietà diverse, e usano meccanismi di routing per modellare relazioni parte-tutto.

L'apprendimento di embeddings con pochi esempi (few-shot learning) o zero esempi (zero-shot learning) affronta la sfida di generalizzare a parole o concetti nuovi con minima o nulla evidenza. Meta-learning permette ai modelli di "imparare a imparare" rappresentazioni efficienti che si adattano rapidamente a nuove parole basandosi su pochi contesti. Questo è cruciale per lingue a basse risorse o domini specializzati dove i dati sono scarsi.

Gli embeddings personalizzati adattati a utenti individuali o comunità specifiche riconoscono che il linguaggio varia tra parlanti. Dialetti, idioletti, registri professionali creano variabilità lessicale. Embeddings personalizzati potrebbero catturare queste differenze, migliorando sistemi conversazionali o di raccomandazione che devono adattarsi a stili individuali.

Le tecniche di compression e quantizzazione mirano a ridurre il footprint di memoria e computazionale degli embeddings senza degradare significativamente la qualità. Product quantization decompone vettori in sotto-vettori e quantizza ciascuno indipendentemente. Binarizzazione riduce embeddings a vettori binari. Queste tecniche sono cruciali per deployment su dispositivi mobili o edge con risorse limitate.

L'interpretabilità è una priorità crescente. Tecniche come sparse coding, disentanglement (separazione di fattori di variazione), e attribute-based embeddings cercano di rendere le dimensioni degli embeddings più interpretabili. Se potessimo identificare che una dimensione codifica "animatezza", un'altra "concretezza", un'altra "polarità emotiva", questo faciliterebbe debugging, analisi, e applicazioni che richiedono reasoning esplicito.

Gli embeddings fairness-aware incorporano obiettivi di equità durante l'addestramento, non solo come post-processing. Adversarial training, re-weighting di esempi, o modifiche alla funzione obiettivo possono produrre embeddings che sono allo stesso tempo performanti e meno distorti rispetto ad attributi protetti. Definizioni multiple di fairness (demographic parity, equalized odds, individual fairness) possono essere operazionalizzate e ottimizzate.

L'esplorazione di architetture neurali innovative continua. I Transformer hanno dominato recentemente, ma architetture alternative come State Space Models (S4), modelli basati su graph neural networks, o approcci neurosimbolici che integrano reasoning simbolico con rappresentazioni neurali potrebbero offrire vantaggi complementari.

Conclusioni: La Geometria come Finestra sul Significato

Gli spazi vettoriali semantici hanno rivoluzionato il modo in cui le macchine rappresentano e processano il linguaggio, trasformando parole da simboli discreti e atomici in punti in spazi geometrici continui dove la posizione relativa codifica relazioni semantiche complesse. Questa rappresentazione geometrica non è una mera tecnica computazionale ma riflette principi profondi dell'organizzazione del significato linguistico.

L'emergere di pattern geometrici regolari - analogie come offset paralleli, sinonimi come cluster densi, antonimi come direzioni opposte - da processi di apprendimento puramente statistico è notevole. Questi pattern non sono programmati esplicitamente ma emergono dall'analisi delle regolarità distribuzionali del linguaggio in uso. Questo fenomeno supporta l'ipotesi distribuzionale e suggerisce che molto della struttura semantica è accessibile attraverso l'analisi di come le parole sono usate in contesto.

La transizione dagli embeddings statici come Word2Vec agli embeddings contestualizzati come BERT rappresenta una progressione verso rappresentazioni più ricche e flessibili che catturano la natura dinamica e contestualmente dipendente del significato. Tuttavia, anche questi modelli avanzati hanno limitazioni: l'assenza di grounding percettivo, la difficoltà con ragionamento logico complesso, la sensibilità a bias nei dati.

La scoperta e l'analisi di bias negli embeddings ha sollevato importanti questioni etiche e sociali. Gli embeddings non sono specchi neutrali del linguaggio ma assorbono e possono amplificare pregiudizi sociali. Affrontare questi problemi richiede non solo soluzioni tecniche ma anche consapevolezza critica, trasparenza, e considerazione degli impatti sociali delle tecnologie di AI.

Le applicazioni degli embeddings spaziano dal recupero di informazioni alla traduzione, dalla classificazione alla generazione, dimostrando la loro versatilità e utilità pratica. Allo stesso tempo, la ricerca continua a esplorare direzioni innovative: integrazione multimodale, incorporazione di conoscenza strutturata, modelli probabilistici, architetture neurali avanzate.

Dal punto di vista teorico, gli embeddings offrono spunti affascinanti sulla natura del linguaggio e della cognizione. Dimostrano che rappresentazioni distribuite possono catturare strutture complesse, supportano teorie empiriste dell'acquisizione linguistica, e illuminano dibattiti sulla relazione tra simbolico e subsimbolico, tra sintassi e semantica, tra linguaggio e pensiero.

La geometria del significato rivelata dagli embeddings è una finestra su come il linguaggio organizza e struttura la conoscenza. Gli spazi ad alta dimensionalità dove le parole risiedono sono abitati non casualmente ma secondo principi regolari che riflettono l'uso comunicativo, le relazioni concettuali, e le strutture tassonomiche. Esplorare questi spazi, comprenderne la topologia e la geometria, significa comprendere aspetti fondamentali di come gli esseri umani categorizzano l'esperienza, comunicano pensieri, e costruiscono significati condivisi.

Le limitazioni attuali degli embeddings non dovrebbero oscurare i progressi straordinari compiuti. In pochi decenni, siamo passati da rappresentazioni simboliche discrete che

trattavano le parole come etichette opache a rappresentazioni geometriche ricche che catturano sfumature semantiche, relazioni analogiche, e organizzazioni categoriali. Questo progresso è stato possibile grazie alla confluenza di intuizioni teoriche (ipotesi distribuzionale, rappresentazioni distribuite), progressi algoritmici (reti neurali, ottimizzazione efficiente), e disponibilità di dati e computazione su larga scala.

Guardando al futuro, è probabile che le rappresentazioni semantiche diventeranno ancora più ricche e sfaccettate, integrando informazioni multimodali, conoscenza strutturata, e capacità di ragionamento. La ricerca continuerà a esplorare sia il miglioramento delle prestazioni empiriche sia la comprensione teorica di cosa rende efficaci queste rappresentazioni. Le questioni etiche e sociali rimarranno centrali, richiedendo impegno continuo per sviluppare tecnologie che siano non solo potenti ma anche eque e responsabili.

La geometria del significato negli spazi vettoriali semantici è più di una curiosità matematica o una tecnica ingegneristica: è un paradigma che ha trasformato il processamento del linguaggio naturale e ha aperto nuove prospettive sulla natura del significato stesso. Che si tratti di migliorare motori di ricerca, di permettere a macchine di tradurre tra lingue, di assistere nella comunicazione umana, o di sondare domande filosofiche sulla semantica, gli spazi vettoriali continuano a essere strumenti potenti e fonti di insight.

In conclusione, gli word embeddings esemplificano come concetti astratti - il significato, le relazioni semantiche, la struttura concettuale - possano essere formalizzati, misurati, e manipolati attraverso il linguaggio della matematica e della geometria. Dimostrano che la distinzione tra scienze umanistiche e scienze formali non è assoluta: lo studio del linguaggio, fenomeno quintessenzialmente umano, può beneficiare profondamente da strumenti quantitativi, e viceversa, lo sviluppo di tecnologie avanzate richiede comprensione profonda delle sfumature linguistiche e sociali. Questa intersezione disciplinare, questa fusione di quantitativo e qualitativo, di formale e interpretativo, è forse una delle lezioni più preziose che gli spazi vettoriali semantici ci insegnano.

Riferimenti Fondamentali

Sebbene questo paper sia stato sviluppato come trattazione sintetica e integrata, le idee e i risultati discussi si basano su decenni di ricerca in linguistica computazionale, machine learning, e scienze cognitive. Di seguito indichiamo alcune delle pubblicazioni e contributi fondamentali che hanno plasmato il campo degli spazi vettoriali semantici.

I lavori pionieristici sull'ipotesi distribuzionale risalgono a Zellig Harris negli anni Cinquanta con la distributional hypothesis e a John Rupert Firth con il celebre aforisma sulle parole e la loro compagnia. Questi contributi linguistici hanno fornito la base teorica per approcci computazionali successivi.

Negli anni Novanta, i modelli di linguaggio neurali hanno iniziato a emergere con il lavoro di Yoshua Bengio e colleghi sulla A Neural Probabilistic Language Model, che ha introdotto l'idea di apprendere embeddings come parte dell'addestramento di modelli di linguaggio. La Latent Semantic Analysis di Landauer e Dumais ha dimostrato il potere della fattorizzazione matriciale per catturare semantica da matrici termine-documento.

Il lavoro di Tomas Mikolov e colleghi su Word2Vec, pubblicato in articoli come Efficient Estimation of Word Representations in Vector Space e Distributed Representations of Words and Phrases and their Compositionality, ha democratizzato gli embeddings e dimostrato proprietà come le analogie vettoriali. GloVe è stato introdotto da Pennington, Socher e Manning in GloVe: Global Vectors for Word Representation. FastText è stato sviluppato da Bojanowski e colleghi in Enriching Word Vectors with Subword Information.

Il lavoro su bias negli embeddings di Bolukbasi e colleghi, Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings, ha catalizzato la discussione su equità e responsabilità. Studi successivi di Caliskan, Bryson e Narayanan in Semantics derived automatically from language corpora contain human-like biases hanno ulteriormente documentato la pervasività dei bias.

Gli embeddings contestualizzati sono stati introdotti con ELMo da Peters e colleghi in Deep contextualized word representations, e rivoluzionati da BERT di Devlin e colleghi in BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Il Transformer, architettura alla base di BERT, è stato introdotto da Vaswani e colleghi in Attention is All You Need.

I Poincaré embeddings per catturare gerarchie sono stati proposti da Nickel e Kiela in Poincaré Embeddings for Learning Hierarchical Representations. Il lavoro su interpretabilità e analisi di embeddings include contributi di Levy e Goldberg come Linguistic Regularities in Sparse and Explicit Word Representations e Neural Word Embedding as Implicit Matrix Factorization.

Studi sull'emergere di struttura negli spazi neurali da prospettive neuroscientifiche e cognitive includono lavori su rappresentazioni distribuite di Hinton e colleghi. La topological data analysis applicata agli embeddings è un'area emergente con contributi recenti.

Testi fondamentali che coprono embeddings e NLP più ampiamente includono Speech and Language Processing di Jurafsky e Martin, Neural Network Methods for Natural Language Processing di Yoav Goldberg, e Introduction to Information Retrieval di Manning, Raghavan e Schütze.

Questa bibliografia rappresenta solo una piccola frazione della letteratura vasta e in rapida crescita sugli spazi vettoriali semantici. Il campo continua a evolversi rapidamente, con nuove architetture, metodi, e applicazioni che emergono costantemente. L'intersezione di linguistica, informatica, matematica e scienze cognitive che caratterizza questa area di ricerca promette sviluppi continui e insight profondi sulla natura del linguaggio e del significato.

L'esplorazione di come relazioni semantiche complesse - analogie, sinonimia, antinomia, gerarchie tassonomiche - emergano naturalmente dalla rappresentazione geometrica in spazi ad alta dimensionalità rivela principi organizzativi profondi del linguaggio. Questi principi non sono imposti dall'alto ma emergono bottom-up dall'analisi statistica di vasti corpus testuali, dimostrando la potenza dell'apprendimento basato sui dati.

Tuttavia, è essenziale mantenere una prospettiva critica e riconoscere che le rappresentazioni computazionali, per quanto sofisticate, catturano solo aspetti del significato

umano. Il linguaggio è profondamente intrecciato con l'esperienza corporea, l'interazione sociale, la cultura, la creatività - dimensioni che trascendono la pura distribuzione statistica delle parole. Gli embeddings sono strumenti potenti, ma non sono la risposta completa alla domanda "cos'è il significato?".

L'integrazione futura di modalità diverse, di conoscenza strutturata, di capacità di ragionamento, e di considerazioni etiche promette di portare le rappresentazioni semantiche verso maggiore ricchezza e responsabilità. Il viaggio verso la comprensione computazionale profonda del linguaggio è ancora in corso, e gli spazi vettoriali semantici rappresentano una tappa significativa ma non finale in questo percorso.

Concludiamo questo saggio con l'osservazione che la geometria, disciplina antica della matematica, trova una nuova e inaspettata applicazione nello studio del significato linguistico. Le forme e le strutture che Euclide esplorò nei suoi Elementi risuonano, trasformate, negli spazi ad alta dimensionalità dove le parole danzano secondo pattern che riflettono secoli di uso umano. Questa connessione profonda tra forma e significato, tra spazio e simbolo, tra quantità e qualità, è testimonianza dell'unità fondamentale della conoscenza umana e della bellezza che emerge quando discipline diverse convergono su problemi comuni.